Lecture Notes in Computer Science 2742
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Rebecca N. Wright (Ed.)

Financial Cryptography

7th International Conference, FC 2003
Guadeloupe, French West Indies, January 27-30, 2003
Revised Papers

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Rebecca N. Wright
Stevens Institute of Technology, Department of Computer Science
Castle Point on Hudson, Hoboken, NJ 07030, USA
E-mail: rwright@cs.stevens-tech.edu

Cataloging-in-Publication Data applied for

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): E.3, D.4.6, K.6.5, K.4.4, C.2, J.1, F.2.1-2

ISSN 0302-9743
ISBN 3-540-40663-8 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

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

http://www.springer.de

© Springer-Verlag Berlin Heidelberg 2003
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin GmbH
Printed on acid-free paper SPIN: 10929216 06/3142 5 4 3 2 1 0

Preface

The 7th Annual Financial Cryptography Conference was held January 27–30, 2003, in Guadeloupe, French West Indies. Financial Cryptography is organized by the International Financial Cryptography Association.

Financial Cryptography 2003 received 54 paper submissions, of which one was withdrawn. The remaining papers were carefully reviewed by at least three members of the program committee. The program committee selected 17 papers for inclusion in the conference, revised versions of which are included in this proceedings.

In addition to the submitted papers, the program included interesting and entertaining invited talks by Tim Jones on digital cash and by Richard Field on the interactions between technology and the United Nations. There were also several panels, on micropayments, economics of security, and trusted computing platforms, some of which are represented by contributions in these proceedings, and a rump session chaired by Juan Garay.

We thank the program committee (listed on the next page) for their hard work in selecting the program from these papers. We also thank the external referees who helped with the reviewing task: N. Asokan, Danny Bickson, Emmanuel Bresson, Dario Catalano, Xuhua Ding, Louis Granboulan, Stuart Haber, Amir Herzberg, Bill Horne, Russ Housley, Yongdae Kim, Brian LaMacchia, Phil MacKenzie, Maithili Narasimha, Phong Nguyen, Kaisa Nyberg, David Pointcheval, Tomas Sander, Yaron Sella, Mike Szydlo, Anat Talmy, Ahmed Tewfik, Susanne Wetzel, Shouhuai Xu, and Jeong Yi. (Apologies for any omissions inadvertent.)

We thank Phong Nguyen and David Pointcheval for all their work as general chairs. We thank the conference sponsors (listed on the next page) for their financial support. Thanks also to Thomas Herlea for running the WebReview system that was used for the electronic reviewing of the submitted papers.

Finally, we thank all authors who submitted papers, and all conference attendees. Without them, there would have been no conference.

May 2003 Rebecca N. Wright

Program Chairs

Jean Camp	Harvard University
Rebecca Wright	Stevens Institute of Technology

Program Committee

Chris Avery	Harvard University
Dan Burk	University of Minnesota
Lorrie Cranor	AT&T Labs
Carl Ellison	Intel Labs
Ian Goldberg	Zero Knowledge Systems
John Ioannidis	AT&T Labs
Markus Jakobsson	RSA Laboratories
Ari Juels	RSA Laboratories
Helger Lipmaa	Helsinki University of Technology
Dahlia Malkhi	Hebrew University of Jerusalem
Satoshi Obana	NEC
Andrew Odlyzko	University of Minnesota
Benny Pinkas	HP Labs
Jacques Stern	École Normale Supérieure
Gene Tsudik	University of California at Irvine

General Chairs

Phong Nguyen	École Normale Supérieure
David Pointcheval	École Normale Supérieure

Sponsors

Silver Sponsor:	nCipher
Bronze Sponsors:	France Telecom R&D, RSA Security
In-Kind Sponsor:	CNRS

Financial Cryptography 2003 was organized by the International Financial Cryptography Association.

Table of Contents

Micropayment and E-cash

Using Trust Management to Support Transferable Hash-Based Micropayments 1
Simon N. Foley

A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks 15
Markus Jakobsson, Jean-Pierre Hubaux, Levente Buttyán

On the Anonymity of Fair Offline E-cash Systems 34
Matthieu Gaud, Jacques Traoré

Retrofitting Fairness on the Original RSA-Based E-cash 51
Shouhuai Xu, Moti Yung

Panel: Does Anyone Really Need MicroPayments?

Does Anyone Really Need MicroPayments? 69
Nicko van Someren, Andrew Odlyzko, Ron Rivest, Tim Jones, Duncan Goldie-Scot

The Case Against Micropayments 77
Andrew Odlyzko

Security, Anonymity, and Privacy

On the Economics of Anonymity 84
Alessandro Acquisti, Roger Dingledine, Paul Syverson

Squealing Euros: Privacy Protection in RFID-Enabled Banknotes 103
Ari Juels, Ravikanth Pappu

How Much Security Is Enough to Stop a Thief? 122
Stuart E. Schechter, Michael D. Smith

Attacks

Cryptanalysis of the OTM Signature Scheme from FC'02 138
Jacques Stern, Julien P. Stern

"Man in the Middle" Attacks on Bluetooth 149
Dennis Kügler

Fault Based Cryptanalysis of the Advanced Encryption Standard (AES) 162
Johannes Blömer, Jean-Pierre Seifert

Panel: Economics of Security

Economics, Psychology, and Sociology of Security 182
Andrew Odlyzko

Fair Exchange

Timed Fair Exchange of Standard Signatures 190
Juan A. Garay, Carl Pomerance

Asynchronous Optimistic Fair Exchange Based on Revocable Items 208
Holger Vogt

Auctions

Fully Private Auctions in a Constant Number of Rounds 223
Felix Brandt

Secure Generalized Vickrey Auction Using Homomorphic Encryption 239
Koutarou Suzuki, Makoto Yokoo

Panel: Trusted Computing Platforms

Trusted Computing Platforms: The Good, the Bad, and the Ugly 250
Moti Yung

On TCPA 255
Dirk Kuhlmann

Cryptographic Tools and Primitives

On The Computation-Storage Trade-Offs of Hash Chain Traversal 270
Yaron Sella

Verifiable Secret Sharing for General Access Structures, with Application to Fully Distributed Proxy Signatures 286
Javier Herranz, Germán Sáez

Non-interactive Zero-Sharing with Applications to Private Distributed Decision Making 303
Aggelos Kiayias, Moti Yung

Author Index 321

Using Trust Management to Support Transferable Hash-Based Micropayments

Simon N. Foley

Department of Computer Science,
University College, Cork, Ireland
s.foley@cs.ucc.ie

Abstract. A hash-chain based micropayment scheme is cast within a trust management framework. Cryptographic delegation credentials are used to manage the transfer of micropayment contracts between public keys. Micropayments can be efficiently generated and determining whether a contract and/or micropayment should be trusted (accepted) can be described in terms of a trust management compliance check. A consequence is that it becomes possible to consider authorisation based, in part, on monetary concerns. The KeyNote trust management system is used to illustrate the approach.

Keywords: Delegation; Digital Cash; One-way Hash Functions; Public Key Certificates; Trust Management.

1 Introduction

Trust Management [3,4,9,21] is an approach to constructing and interpreting the trust relationships between public keys that are used to mediate security critical actions. Cryptographic credentials are used to specify delegation of authorisation among public keys.

In this paper we consider how Trust Management can be used to manage trust relationships that are based on monetary payment. A benefit to characterising a payment scheme as a trust management problem is the potential to support, within the trust management framework, sophisticated trust requirements that can combine monetary and authorisation concerns. For example, authorisation to access a valuable resource might be based on some suitable combination of permission and monetary payment or deposit (possibly partially refundable if it can be determined that the resource is not misused). In this case, perhaps a user with 'less' permissions would be required to provide a larger deposit, while a user with 'more' permissions provides a smaller deposit. The trust management system is expected to help the application system manage what is meant by trusted access to this resource.

In [7,14], the KeyNote trust management system is used to manage trust for a micro-billing based payment scheme. Their scheme is similar to IBM's mini-pay scheme [13]: KeyNote is used by the payer (merchant) to determine whether or not an off-line payment from a particular payee (customer) should

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 1–14, 2003.
© Springer-Verlag Berlin Heidelberg 2003

be trusted, or whether the payee should go online to validate the payment and payee. The scheme is intended for small value payments (under $1.00). Since the generation of each payment transaction requires a public key cryptographic operation (signature) then it may not be practical for very low value payments where the cost of processing is high relative to the value of the payment.

This paper extends earlier work [11] exploring how KeyNote can be used to to support low-value micropayments in the provision of authorised access to resources by the participants of a meta-computer The meta-computer [10, 18] is a network of heterogenous computers that work together to solve large problems. It encourages the sharing of resources between different organisations, departments and individuals. A consequence of this sharing is that the providers of the resources expect payment for their computation processing and services, on a per-use basis. For example, participants might be paid for contributing processing power to weather model computation.

In [11], we describe a preliminary KeyNote implementation of micropayments that is based on hash-chains [1,8,20]. These schemes are limited in that it is not possible for the participants to transfer (delegate) their micropayment contracts to third parties. Some form of efficient transfer is desirable. For example, an overloaded client workstation might like to transfer some of its workload and the corresponding micropayment contract that it holds to another client.

The contribution in this paper is the development of a hash-chain based micropayment scheme that is cast within a trust management framework and supporting the efficient transfer of micropayment contracts. Codifying a micropayment scheme in terms of KeyNote also demonstrates the usefulness and applicability of trust management systems in general.

The paper is organised as follows. Section 2 describes a simple model of cryptographic credential based delegation that forms the basis of trust management. Section 3 extends this model by considering how values (that represent permissions) along a hash-chain can be delegated without having to sign new credentials each time. This extension forms the basis of the transferable micropayment scheme that is proposed in Section 4. Section 5 considers how the scheme can be codified and interpreted within the KeyNote trust management system.

2 Delegating Authorisation

A simple model is used to represent delegation of authorisation between public keys. A signed cryptographic credential, represented as $\{\!|\, K_B, p \,|\!\}_{sK_A}$, indicates that K_A delegates to K_B, the authorisation permission p. Permissions are structured in terms of lattice $(PERM, \leq, \sqcap)$, whereby $p \leq q$ means that permission q provides no less authorisation than p. A simple example is the power-set lattice of $\{\mathsf{read}, \mathsf{write}\}$, with ordering defined by subset, and greatest lower bound ($\sqcap$) defined by intersection.

Given a credential $\{\!|\, K_B, q \,|\!\}_{sK_A}$ and $p \leq q$ then there is an implicit delegation of p to K_B by K_A, written as $(\!|\, K_B, p \,|\!)_{K_A}$. Two reduction rules follow.

$$\frac{\{\!|\, K_B, p \,|\!\}_{sK_A}}{(\!|\, K_B, p \,|\!)_{K_A}} \text{ [R1]} \qquad \frac{(\!|\, K_B, p \,|\!)_{K_A}; p' \leq p}{(\!|\, K_B, p' \,|\!)_{K_A}} \text{ [R2]}$$

If delegation is regarded as transitive, and if K_A delegates p to K_B, and K_B delegates p to K_C, then it follows that K_A implicitly delegates p to K_C.

$$\frac{(\!|\, K_C, p \,|\!)_{K_B}; (\!|\, K_B, p' \,|\!)_{K_A}}{(\!|\, K_C, p \sqcap p' \,|\!)_{K_A}} \text{ [R3]}$$

This corresponds to SPKI certificate reduction [9] (greatest lower bound is equivalent to SPKI tuple intersection). It is not unlike a partial evaluation over a collection of KeyNote credentials, resulting in a single equivalent credential. Such reduction can be efficiently implemented using a depth first search of a delegation graph such as [2,6]. At this point, we do not consider permissions that cannot be further delegated and nor do we consider threshold schemes.

These rules are used by a trust management system to determine whether a request for an action (permission) is authorised. If the local policy specifies that K_A is authorised for p, and delegation $(\!|\, K_B, p \,|\!)_{K_A}$ can be deduced from some collection of credentials, then it follows that K_B is authorised for p.

3 Delegating Hash Chain Values

Consider the set of permission values *PERM* that are drawn from the range of a one-way cryptographic hash function $h()$. Section 4 provides one possible interpretation for such permissions that are based on hash-chain micropayments whereby each permission (a hash value micropayment) represents authorisation for payment.

Example 1 Principal K_A securely generates a secret random seed s and generates credential $\{\!|\, K_B, h(s) \,|\!\}_{sK_A}$, delegating the 'permission' value $h(s)$ to K_B. With this credential, K_B is authorised for permission $h(s)$, but is not authorised for the permission s.

Suppose that at some later point in time, K_A wishes to delegate the permission s to K_B. Rather than having to sign a new credential $\{\!|\, K_B, s \,|\!\}_{sK_A}$, K_A sends the value s over an unsecured channel to K_B. Principal K_B can then prove that it has been authorised by K_A for the permission s by presenting the original credential $\{\!|\, K_B, h(s) \,|\!\}_{sK_A}$ and the value s: a third party simply checks that the hash of the s presented compares with the permission for which it holds a credential. The one-way nature of the hash function ensures that it is not feasible for K_B to forge the s permission before it is revealed. △

In general, a delegator K_A generates a secure random seed s and computes and issues credential $\{\!|\, K_B, h^n(s) \,|\!\}_{sK_A}$. If the seed s is known only to K_A then $[h^{n-1}(s), \ldots h^0(s)]$ forms a totally ordered chain of permissions that can be delegated to K_B by revealing the corresponding hash-value, one at a time, in order. Principal K_B is authorised for the permission x if $h^i(x)$ matches the value $h^n(s)$

in the original credential for some i known by the principal. We assume that the delegator K_A maintains a suitable interpretation for the semantics of the permissions it issues. The advantage of taking this approach is that K_A need sign only one initial delegation credential; subsequent delegations (along the permission chain, starting at the credential) do not require further costly cryptographic signatures. Since $h()$ is a one-way cryptographically secure hash function then it is not possible for the delegatee K_B to forge or to compute a permission that is yet to be delegated from the permission hash chain.

We provide an interpretation for this *hash-chain delegation* in terms of our simple model of delegation. This is done by defining an ordering relation over hash-value permissions $PERM$ as follows.

Definition 1 For $x, y \in PERM$ then $x \leq y$, with respect to a some principle, if and only if the principal knowing y can *feasibly determine* some x and i such that $h^i(x) = y$. △

Note that $x \leq y$ should not be taken to mean that a principal can reverse the hash function, rather, it means that the value has been revealed by a principal who knows the initial seed. In addition, note that the orderings in this relation increase (monotonically) to reflect what a principal can feasibly compute based on the hash permissions that it has received to date. For Example 1, delegatee K_B is in possession of $\{\!|\, K_B, y \,|\!\}_{sK_A}$ where $y = h(s)$ and cannot feasibly find some x such that $h^i(x) = y$. However, once s has been revealed to K_B then it knows to calculate $h^1(s) = y$ and thus $s \leq y$.

Applying this definition of permission ordering to the certificate reduction rule R2 illustrates how revealing a hash permission generates an implicit delegation.

$$\frac{(\!|\, K_B, y \,|\!)_{K_A}; \text{can feasibly compute } h(x) = y}{(\!|\, K_B, x \,|\!)_{K_A}}$$

Example 2 Suppose that K_A generates a secret seed s and computes permissions $a = h^4(s)$, $b = h^3(s)$, $c = h^2(s)$ and $d = h^1(s)$. K_A first delegates permission a to K_B by writing $\{\!|\, K_B, a \,|\!\}_{sK_A}$ and then reveals b and c (see Figure 1).

K_B can prove that it is authorised by K_A for permission c since it is authorised for a and it can feasibly compute $h(c) = b$ and and $h(b) = a$ and therefore, $c \leq b \leq a$, and by reduction $(\!|\, K_B, c \,|\!)_{K_A}$.

Suppose that K_B delegates authorisation for c to K_C by writing credential $\{\!|\, K_C, c \,|\!\}_{sK_B}$. K_C can prove that it is authorised by K_A for c as follows. K_C holds credential chain $\{\!|\, K_B, a \,|\!\}_{sK_A}, \{\!|\, K_C, c \,|\!\}_{sK_B}$. Since it knows permission c it can feasibly calculate permissions $b = h(c)$ and $a = h(b)$ and therefore $c \leq b \leq a$. Thus, reduction rule R2 applied to $\{\!|\, K_B, a \,|\!\}_{sK_A}$ gives implicit delegation $(\!|\, K_B, c \,|\!)_{K_A}$ and this, with credential $\{\!|\, K_C, c \,|\!\}_{sK_B}$ gives $(\!|\, K_C, c \,|\!)_{K_B}$ by reduction rule R3, that is, K_C is authorised for c by K_A.

Note that in this case it is not possible to use the reduction rules to prove that K_C is authorised for permissions a and b. To become authorised for these

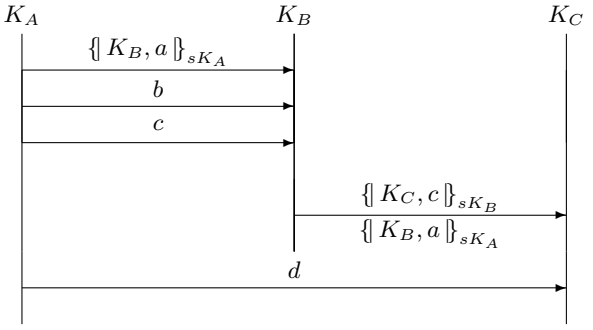

Fig. 1. Hash-Chain Delegation

permissions it is necessary for K_B (or K_A) to explicitly delegate the permission by signing a suitable initial credential such as $\{\!|\, K_C, a \,|\!\}_{sK_B}$.

K_A delegates permission d by making the value public, whereupon it becomes possible to deduce $(\!|\, K_C, d \,|\!)_{K_A}$ and $(\!|\, K_B, d \,|\!)_{K_A}$. △

This scheme is not limited to a single secret seed s: principals may generate and use as many seeds as they require. Assuming that seeds are generated in a cryptographically secure manner then the properties of the one-way hash function will ensure that collisions between permission chains are unlikely and that permission forgery is infeasible. When multiple seeds are used the permission ordering is composed of a series of independent chains (total orderings that can be feasibly generated) from each seed. In this case, the greatest lower bound operation $a \sqcap b$ is the lower of a and b when they are comparable (same chain). If they are incomparable (different chain) then the result of the operation is lattice 'bottom'.

If the manner of delegation is such that it can be structured according to a collection of hash-chains, as above, then we have an efficient way of performing delegation. Once the initial delegation credential $\{\!|\, K_B, y \,|\!\}_{sK_A}$ is signed and issued then subsequent delegation of permissions along that chain do not require costly cryptographic signatures. The next section considers transferable hash-chain micropayments as a practical application of this general approach. Whether other applications of hash-chain delegation exist is a topic for future research.

In [16,17] hash-chains are used to provide an efficient method for revocation of public key certificates. It can be interpreted within our framework as follows. With certificate $\{\!|\, K_B, [p, n, h^n(s)] \,|\!\}_{sK_A}$, K_A delegates potential authorisation for permission p to K_B for n time-periods. Each hash value along the hash chain represents an authorisation for a particular time-period (starting with $h^n(s)$ for time period 0) In this case, K_A is considered authorised for time period i if a hash value v is presented such that $h^i(v) = h^n(s)$. At the start of a new time period, the certificate issuer makes available the corresponding hash value. Deciding not to issue a hash-value provides an efficient form of revocation (at the granularity of the time-period). In [16,17] a hash-chain gives a authorisation time-line for a

single permission. This differs slightly to our interpretation, where a hash-chain represents a particular total ordering over a set of permissions.

4 Transferable Hash Chain Micropayments

Hash-based micropayments schemes are intended to support very low-value payments and operate as follows.

A *payer* (the principal making the payment) securely generates a fresh random seed s, and computes $h^n(s)$, where $h()$ is a cryptographic one-way hash function. If s is known only to the payer, then $[h^{n-1}(s), n-1, val] \ldots [h^1(s), 1, val]$ provides an ordered chain of micropayments, each one worth val. Initially, the payer provides a *payee* with $[h^n(s), n, val]$, which acts as a contract for $(n-1)$ micropayments. It is required that the payer is unforgeably linked with the contract; for example, the payer signs the contract.

A payee (the principal receiving the payment) who has securely received i micropayments, $[h^{n-1}(s), n-1, val] \ldots [h^{n-i}(s), n-i, val]$, can use the hash function $h()$ to check their validity against the initial contract. Since $h()$ is a one-way hash function then it is not feasible for the payee to forge or compute the next $(i+1)^{th}$ payment (before it is issued). Micropayments may be cashed in/reimbursed by the payee at any time: the payment plus contract provides irrefutable evidence to a third party of the payer's obligation to honour the payment. To guard against double spending, the payer must keep track of irrefutable evidence of the reimbursements made to a payee against a contract.

This approach to micropayments has been proposed and used in payment schemes proposed by [1,8,20]. For example, in [1], the payer threads digital coins (issued by a bank) through the hash chain such that each micropayment reveals an authentic digital coin that can be reimbursed by the original bank.

In this paper we show how these schemes can, in general, be supported within a trust management system. Taking such an approach allows us to extend existing micropayment schemes by providing a framework that supports the transfer/delegation of micropayment contracts between principals.

Micropayments are interpreted in terms of trust management as follows. A payer sets up a contract by writing a suitable delegation certificate that contains the initial contract hash value. Hash-value micropayments correspond to (hash-value) permissions. A payee can cash in a micropayment if the payee has been (hash-chain) delegated its corresponding permission; this test can be done as a trust management compliance check.

Example 3 Consider Example 2. The payer K_A issues a new contract credential $\{\!|\, K_B, a \,|\!\}_{sK_A}$. For the sake of simplicity, we assume that micropayment value (val) and the length of the hash chain (n) are universally agreed and fixed beforehand.

Two micropayments are made by K_A to K_B as b and c. Payee K_B can confirm that they are valid payments by carrying out a certificate reduction to test $(\!|\, K_B, b \,|\!)_{K_A}$, and so forth. Payer K_A can validate a claim for reimbursement

of a micropayment b from K_B by testing that K_B can prove $(\!| K_B, b |\!)_{K_A}$. We assume that K_A also checks for double spending. △

The proposed scheme allows a principal to transfer part of a micropayment contract to a third party. Suppose that a principal K_B holds a contract credential $\{\!| K_B, x |\!\}_{sK_A}$ and has been paid up to micropayment y, where $x = h^i(y)$ and, therefore, implicitly holds $(\!| K_B, y |\!)_{K_A}$. Principal K_B can transfer the remainder of this contract to a third party K_C by signing $\{\!| K_C, y |\!\}_{sK_A}$. In signing this credential, K_C is declaring that it gives up any claim that it originally held to seek reimbursement from K_A for micropayments subsequent to y, based on its original contract. The recipient K_C uses credential chain $[\{\!| K_B, x |\!\}_{sK_A}, \{\!| K_C, y |\!\}_{sK_B}]$ as a contract for any claims for reimbursement from K_A for micropayments subsequent to y.

Example 4 Continuing Example 3 (Figure 1), K_B transfers the remainder of its micropayment contract with K_A to K_C by writing $\{\!| K_C, c |\!\}_{sK_B}$. K_C later claims for reimbursement of micropayment d from K_A by providing certificate chain $[\{\!| K_B, a |\!\}_{sK_A}, \{\!| K_C, c |\!\}_{sK_B}]$ as evidence of a valid contract along with d. As before, the validity of this claim is tested as a trust management compliance check for $(\!| K_C, d |\!)_{K_A}$. △

Note that the proposed scheme can be used to provide information to assist in the resolution of double payment on transferred contracts. Consider Example 4:

- If no claim for reimbursement for d has yet been made to K_A then that claim can be met.
- If K_A has already reimbursed K_C up to d and subsequently receives a claim for d from K_B then it will reject the claim. In this case it can provide the credential chain $[\{\!| K_B, a |\!\}_{sK_A}, \{\!| K_C, c |\!\}_{sK_B}]$ on the original claim to prove that K_B has given up the right to make such a claim.
- If K_A has already reimbursed K_B and it receives a claim from K_C then it will reject the claim. However it can provide $\{\!| K_C, d |\!\}_{sK_B}$ to prove, in effect, that K_B had agreed not to make such a claim. In this case K_A or K_B may use this credential to seek restitution from K_B.

This scheme may be used to manage simple deposits, whereby a deposit is returned by transferring its original (unused) contract to the issuer. For example, K_A transfers an i micropayment deposit to K_B by signing contract $\{\!| K_B, h^n(s) |\!\}_{sK_A}$ and revealing $h^{n-i}(s)$ (from which the chain of micropayments $h^{n-1}(s) \ldots h^{n-i}(s)$ may be calculated). The deposit recipient may return the deposit by signing $\{\!| K_A, h^n(s) |\!\}_{sK_B}$ and returning it to K_A. As above, this credential provides evidence that K_B does not expect reimbursement for the i micropayments.

5 Towards a KeyNote Based Implementation

In this section we illustrate how the above scheme can be implemented within the KeyNote trust management system [3]. Note that we assume a version of

KeyNote with a minor modification, whereby we assume the ability to invoke an MD5 hash function within the KeyNote credential. Adding such functionality to the KeyNote Standard [3] would be trivial and would not impact on its architecture in any way.

We use the following general strategy. Suppose that K_A holds the top $h^i(s)$ of the hash chain

$$[h^i(s), h^{i-1}(s), \ldots h^{\texttt{ChainInd}}(s), \ldots h^0(s)]$$

K_A may have generated this hash chain (knows the seed s). Alternatively, $k^i(s)$ may be the most recent payment under some contract with another principle (public key) (who knows s) that K_A trusts. In this second case, 'trust' means that K_A trusts the contract issuer for the purposes of the particular contract.

If K_A issues $h^i(s)$ as a contract to K_B then it, in effect, authorises K_B for any hash value `HashVal` (that K_B can feasibly produce) in position `ChainInd` in the above chain such that

$$h^{i-\texttt{ChainInd}}(\texttt{HashVal}) = h^i(s) \tag{1}$$

holds. This authorisation is achieved in KeyNote by K_A writing a credential that conditionally delegates any value of `HashVal` such that the above holds, to K_B. Given particular values for the chain's length i and the chain's top $h^i(s)$, then K_A (delegating to K_B) signs a credential that includes the condition (1) above, defined over attribute variables `ChainInd` and `HashVal`. In this case, K_B must know how to produce suitable values for `ChainInd` and `HashVal` such that the above holds (for a compliance check that K_B is authorised for the given `ChainInd` and `HashVal` to hold). Thus, the compliance check proves the validity of a payment [`ChainInd`,`HashVal`]. K_B cannot forge a payment since to do so would require reversing the one-way hash function.

Suppose that K_B has received j payments and wishes to delegate its remaining contract $h^{(i-j)}(s)$ to K_C. Given particular values for $h^{(i-j)}(s)$ and i and j then K_B (delegating to K_C) signs a credential that includes a condition

$$h^{(i-j)-\texttt{ChainInd}}(\texttt{HashVal}) = h^{(i-j)}(s) \tag{2}$$

defined over attribute variables `ChainInd` and `HashVal`. In this case, K_B must know how to produce suitable values for `ChainInd` and `HashVal` such that (2) above holds. If such values are known (by K_C), then K_C is authorised for [`ChainInd`,`HashVal`] by K_B. From (2), it follows that

$$\begin{aligned} h^j(h^{(i-j)-\texttt{ChainInd}}(\texttt{HashVal})) &= h^j(h^{(i-j)}(s)) \\ \Rightarrow h^{i-\texttt{ChainInd}}(\texttt{HashVal}) &= h^i(s) \end{aligned}$$

and, therefore, if equation (1) above holds in the credential issued by K_A to K_B, then K_C is also authorised for [`ChainInd`,`HashVal`] by K_A: a valid delegation chain exists from K_A to K_C). This argument extends to any length delegation chain that is constructed in this manner. This general strategy is illustrated in the following examples.

We first consider how principals use KeyNote to determine whether they should trust (accept) contracts and subsequent micropayments from third parties.

Example 5 Consider Example 3. Principal "Kb" (Bob) trusts that "Ka" (Alice) will honour individual contracts that are worth up $5.00. This is expressed in terms of the following KeyNote credential.

```
Authorizer: "POLICY"
Licensees: "Ka"
Conditions: @PayVal * @ChainLen <= 500;
```

This *policy* credential defines the conditions under which micropayment contracts from the licensee key "Ka" are trusted by Bob. The conditions are defined using a C-like expression syntax in terms of the *action attributes* of the contract, that is, PayVal and ChainLen. Attribute PayVal gives the value of a single micropayment (in cents) and ChainLen is the length of the contract hash-chain. Note that the prefix '@' on a KeyNote attribute converts the string attribute variable to the integer value it represents. For the purposes of illustration we use names to represent public keys (and do not provide signatures).

Alice sends a request to Bob to accept a contract for up to four micropayments worth one cent each. The details of the contract are described in terms of attributes PayVal and ChainLen, as described above, and the attribute HashVal which gives the contact hash-value in position ChainInd = ChainLen (top of chain). In this case, Alice's contract is described by the following attribute bindings.

PayVal $\leftarrow$ 1;
ChainLen $\leftarrow$ 4;
ChainInd $\leftarrow$ 4;
HashVal $\leftarrow$ "PwhHLtyFH9yPUXEx4StCLA"

Alice signs the following KeyNote credential for this contract.

```
Authorizer: "Ka"
Licensees:  "Kb"
Condition:  @PayVal==1 && @ChainLen==4 &&
  md5(@ChainLen-@ChainInd,HashVal)== "PwhHLtyFH9yPUXEx4StCLA";
Signature: ...
```

Operation md5(i, s) is the i^{th} iteration of the MD5 hash of s, that is, $h^i(s)$. In this credential Ka authorises Kb for any hash value payment HashVal in position ChainInd such that the condition holds. This condition is based on Equation (1) given at the start of Section 5. In the credential above, the contract hash-value is md5$(4, s)$ and is taken to correspond to the value a in Example 3.

Bob uses KeyNote to check whether he should trust this particular contract from Alice: he wishes to determine whether, under his own credential policy above, his key Kb has been suitably authorised by Ka for the attribute bindings

defined above for the contract. Evaluating a KeyNote query for this request given the above credentials returns `true`, indicating a trusted delegation (of contract) from `Ka` to `Kb`.

Suppose that Alice sends the first payment `"uyMmZs0K7qgAKcJ+PAFYLw"`. Bob wishes check that this is an authorised payment, that is, it is a payment for which he can expect to be reimbursed. He verifies that `Kb` is authorised for this proposed payment (`HashVal`) by making a KeyNote query with the following attribute bindings.

`PayVal` $\leftarrow$ 1; (per contract)
`ChainLen` $\leftarrow$ 4; (per contract)
`ChainInd` $\leftarrow$ `3`; (this first payment is in position 3)
`HashVal` $\leftarrow$ "`uyMmZs0K7qgAKcJ` + `PAFYLw`"; (value b from Example 3)

This query returns `true`: Bob now knows another hash value for which `Kb` is authorised by `Ka`.

Alice makes a second payment `"0CJkY1EEisC6OX0S36+jBA"` (corresponding to c in Example 3), which Bob checks by querying KeyNote with action attributes

`PayVal` $\leftarrow$ 1; `ChainLen` $\leftarrow$ 4. `ChainInd` $\leftarrow$ `2`;
`HashVal` $\leftarrow$ "`0CJkY1EEisC6OX0S36` + `jBA`".

and this also evaluates to true. △

Bob now has two hash value micropayments that have been revealed by Alice and are, therefore, implicitly delegated by Alice via the initial contract credential. Bob presents these micropayments and the original contract credential and requests reimbursement from Alice. To determine whether the reimbursement request is valid, Alice must check that the micropayments are authorised based on the original contract credential. This check of validity by Alice is similar to the check carried out by Bob in the example above: given that Alice trusts her own key `Ka` then, a KeyNote query determines whether, given the contract credential, the key `Kb` authorised for the the given micropayments.

Note that KeyNote is used to verify that the micropayment is authorised in principle; it does not consider the potential for double spending, that is, multiple reimbursement requests for the same micropayment. Whether trust management would be useful in managing this is a topic for future research.

The next example illustrates how KeyNote is used to manage the general transfer of partially spent micropayment contracts.

Example 6 Continuing the previous example, suppose that Bob decides to transfer the remainder of his contract with Alice to Clare (`Kc`). As described in Example 4, this is done by Bob signing a credential that delegates authorization for the remaining hash chain of the contract to Clare.

The hash value $h^2(s)$ (hash value designated as c in the example) that Bob currently holds acts as a contract for the remainder of the hash chain that is controlled by Alice. Thus, applying Equation (1) of the general credential writing strategy, Bob signs the following contract credential.

```
Authorizer: "Kb"
Licensees: "Kc"
Condition:
   0<= @ChainInd && @ChainInd <=2 &&
   md5(2-@ChainInd,HashVal)== "0CJkY1EEisC6OX0S36+jBA";
Signature: ....
```

This authorises `Kc` for any `HashVal` (that she can feasibly produce) from the remaining hash chain ($2 \geq$ `ChainInd` ≥ 0).

As was similarly done by Bob, Clare can use KeyNote to check whether she should trust this contract. Assuming that Clare uses the same policy credential as Bob (trusting that `Ka` will honour contracts worth, at most, \$5.00) then she queries KeyNote to ensure that her key `Kc` is authorised for this contract described by attribute bindings:

`PayVal` $\leftarrow 1$; `ChainLen` $\leftarrow 4$. `ChainInd` $\leftarrow 2$;
`HashVal` $\leftarrow$ "`0CJkY1EEisC6OX0S36` + `jBA`".

The credentials of Bob and Clare form a delegation chain from `Ka` to `Kc` authorising the contract hash value and thus the KeyNote query evaluates to `true`.

When Alice makes her third payment `"ICy5YqxZB1uWSwcVLSNLcA"` (corresponding to d in Example 4) uses KeyNote, as done by Bob, to check the validity of this `HashVal` in position `ChainInd=1`. Alice similarly processes a request for reimbursement from Clare by checking that a delegation chain exists from `Ka` to `Kc` for the particular payment. Recall from Section 4, that having made a contract transfer, then it is assumed that Bob is trusted not to claim for reimbursements of subsequent payments. If Alice decides that Bob is not trusted then it is straightforward in KeyNote for Alice to use a credential that authorizes Bob's contract such that the authorization cannot be further delegated/transferred. △

One difficulty with the above KeyNote based scheme is that it generally requires repeated hash calculations when determining the validity of a micropayment. Consequently, the performance advantage of using a payment scheme based on one-way hash functions (instead of a public key based scheme such as [7]) diminishes as payments further down the hash chain are spent.

This performance penalty is avoided if the payment recipient compares, instead, the current hash payment (`HashVal`) against the previous payment (`PrevHash`) by checking that

$$\texttt{PrevHash} == \texttt{md5}(1, \texttt{HashVal})$$

The following example illustrates how this check can be managed by the trust management system.

Example 7 Consider Example 5. Bob trusts that `Ka` will honour contracts worth up to 500 one cent micropayments. Bob offers service actions `X` and `Y`, charging 1 cent and 2 cents each time they are used, respectively. Bob's revised policy credential is specified as follows.

```
Authorizer: "POLICY"
Licensees: "Ka"
Conditions:
  Action=="NewContract" && @PayVal== 1 && @ChainLen<=500;
  Action=="X" -> {PrevHash==md5(1,HashVal) && @ChainInd>=0};
  Action=="Y" -> {PrevHash==md5(2,HashVal) && @ChainInd>=0};
```

Attribute `Action` defines the action requested: `"NewContract"` corresponds to a request to accept a new payment contract, and `"X"` and `"Y"` are the service actions offered.

When Bob receives a `"NewContract"` request from Alice he, as before, uses KeyNote to check that his own key `Kb` is suitably authorised under this policy. If the contract is accepted, then Bob stores the contract hash value as attribute `PrevHash`. When Alice requests service action `X`, Bob uses KeyNote to check that `Ka` is authorised for payment `HashVal`, given `Action` $\leftarrow$ `X`, etc. The compliance check succeeds when the payment upholds policy condition `PrevHash==md5(1,HashVal)` and Bob sets `PrevHash` to `HashVal`.

To use service `Y`, a hash-value that permits calculation of two payments must be presented by the payer. For example, if `PrevHash` is b (Example 2), then the `HashVal` presented is d, corresponding to payments d and $c = h(d)$, and policy condition `PrevHash==md5(2,HashVal)` is satisfied. $\triangle$

6 Discussion and Conclusion

This paper describes a hash-chain based micropayment scheme that is cast within a trust management framework and supports the efficient transfer of micropayment contracts. Cryptographic credentials are used to specify the transfer (delegation) of contracts between public keys. Determining whether a micropayment contract or payment should be trusted (accepted) corresponds to a trust management compliance check. A benefit to characterising the payment scheme in this way is that sophisticated trust requirements can be constructed both in terms of monetary and more conventional authorization concerns: access control can be based, in part, on the ability to pay. Codifying a micropayment scheme using KeyNote also demonstrates the usefulness and applicability of trust management systems in general.

The micro-billing scheme [7] uses KeyNote to help determine whether a micro-check (a KeyNote credential, signed by a customer) should be trusted and accepted as payment by a merchant. In [5] these microchecks provide a convenient way to manage the purchase of the hash-chain based coin stacks that are used in [19] to pay for wireless LAN/IEEE 802.11 access to public infrastructure. We believe that it would be straightforward to apply the technique described in our paper to extend [5] to help (trust) manage the transfer of unspent portions of coin stacks between principals.

Hash-chains are used in [16,17] to provide an authorization time-line for permission and provides an efficient method for issuing fresh public key certificates.

It would be interesting to explore how our proposed strategy for coding hash-chains within KeyNote credentials could be applied to technique in [16,17] to provide support for this technique for KeyNote credentials.

In [12,15], hash functions are used to provide an efficient implementation of authentication. In this paper, hash functions are used to support an efficient form of delegation of authorization. Our proposed hash-chain delegation is restricted to orderings that can be characterised as disjoint collections of total orders (hash-chains) over permissions. Hash-chain micropayments are one example of this type of ordering. We are investigating how other techniques might be used to support a similar form of delegation of more general permissions.

Acknowledgements. Thanks to the anonymous reviewers for their useful comments on this paper and for drawing my attention to [16,17]. Thanks also to Angelos Keromytis for his comments and for providing access to [5] prior to publication. This work was supported, in part, by Enterprise Ireland Informatics Research Initiative.

References

1. Ross Anderson, Harry Manifavas, and Chris Sutherland. Netcard - a practical electronic cash system. In *Cambridge Workshop on Security Protocols*, 1995.
2. Tuomas Aura. Comparison of graph-search algorithms for authorization verification in delegation networks. In *Proceedings of NORDSEC'97*, 1997.
3. M Blaze et al. The keynote trust-management system version 2. September 1999. Internet Request For Comments 2704.
4. M Blaze, J Feigenbaum, and J Lacy. Decentralized trust management. In *Proceedings of the Symposium on Security and Privacy*. IEEE Computer Society Press, 1996.
5. M. Blaze, J. Ioannidis, S. Ionnidis, A. Keromytis P. Nikander, and V. Prevelakis. Tapi: Transactions for accessing public infrastructure. submitted for publication, 2002.
6. Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming*, pages 185–210, 1999.
7. Matt Blaze, John Ioannidis, and Angelos D. Keromytis. Offline micropayments without trusted hardware. In *Financial Cryptography*, Grand Cayman, February 2001.
8. Jean-Paul Boly et al. The ESPRIT project CAFE – high security digital payment systems. In *ESORICS*, pages 217–230, 1994.
9. C Ellison et al. SPKI certificate theory. September 1999. Internet Request for Comments: 2693.
10. Simon N. Foley, Thomas B. Quillinan, and John P. Morrison. Secure component distribution using WebCom. In *Proceeding of the 17th International Conference on Information Security (IFIP/SEC 2002)*, Cairo, Egypt, May 2002.
11. S.N. Foley and T.B Quillinan. Using trust management to support micropayments. In *In proceedings of the Annual Conference on Information Technology and Telecommunications*, Waterford, Ireland, October 2002.

12. Li Gong. Using One-way Functions for Authentication. *Computer Communication Review*, 19(5):8–11, 1989.
13. A. Herzherg and H. Yochai. Mini-pay: Charging per click on the web. In *Sixth International World Wide Web Conference*, 1997.
14. John Ioannidis et al. Fileteller: Paying and getting paid for file storage. In *Proceedings of Financial Cryptography*, March 2002.
15. Philippe A. Janson, Gene Tsudik, and Moti Yung. Scalability and flexibility in authentication services: The kryptoknight approach. In *INFOCOM (2)*, pages 725–736, 1997.
16. S. Micali. Efficient certificate revocation. In *Proceedings of the 1997 RSA Data Security Conference*, 1997.
17. S. Micali. Novomodo: Scalable certificate validation and simplified management. In *Proceedings of the First Annual PKI Research Workshop*, April 2002.
18. J.P. Morrison, D.A. Power, and J.J. Kennedy. A Condensed Graphs Engine to Drive Metacomputing. Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA '99), Las Vagas, Nevada, June 28 - July1, 1999.
19. P. Nikander. Authorization and charging in public WLANs using FreeBSD and 802.1. In *Proceedings of the Annual USENIX Technical Conference, Freenix Track*, 2002.
20. Torben P. Pedersen. Electronic payments of small amounts. In *Security Protocols Workshop*, pages 59–68, 1996.
21. R Rivest and B Lampson. SDSI - a simple distributed security infrastructure. In *DIMACS Workshop on Trust Management in Networks*, 1996.

A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks

Markus Jakobsson[1], Jean-Pierre Hubaux[2], and Levente Buttyán[2⋆]

[1] RSA Laboratories, Bedford, MA 01730, USA. www.markus-jakobsson.com
[2] Laboratory for Computer Communications and Applications, Swiss Federal Institute of Technology, Lausanne, EPFL-IC-LCA, CH-1015 Lausanne, Switzerland.
Jean-Pierre.Hubaux@epfl.ch, buttyan@acm.org

Abstract. We propose a micro-payment scheme for multi-hop cellular networks that encourages collaboration in packet forwarding by letting users benefit from relaying others' packets. At the same time as proposing mechanisms for detecting and rewarding collaboration, we introduce appropriate mechanisms for detecting and punishing various forms of abuse. We show that the resulting scheme – which is exceptionally light-weight – makes collaboration rational and cheating undesirable.

Keywords: Audit, collaboration, detection, micro-payment, rational, routing, multi-hop cellular networks

1 Introduction

Multi-hop cellular networks rely on a set of base stations connected to a backbone network, as in conventional cellular networks (such as GSM), and on the mechanisms of ad hoc networks [20], in which packets are relayed hop by hop between peer wireless stations. The expected benefits of such an approach with respect to conventional cellular networks are multifold. First, the energy consumption of the mobile devices can be reduced. Indeed, the energy consumption required for radio transmission grows super-linearly[1] with the distance at which the signal can be received. Therefore, the battery life of wireless devices can be substantially extended if packets are routed in small hops from the originator to the base station. Second, as an immediate positive side-effect of the reduced transmission energy, interference is reduced. Third, if not too remote from each other, mobile devices can communicate independently from the infrastructure.

⋆ *The second and third author were supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322* (http://www.terminodes.org)

[1] Depending on the setting, the power decay is a function of the distance, ranging typically from the square to the fifth power [5]. The exact function depends on the extent to which the signal is reflected off of buildings, on the nature of the material to be traversed, on the possible interference from other electromagnetic sources, etc.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 15–33, 2003.
© Springer-Verlag Berlin Heidelberg 2003

Fourth, the number of fixed antennas can be reduced; and fifth and finally, the coverage of the network can be increased using such an approach. However, while all participating wireless devices stand to benefit from such a scheme, a cheater could benefit even more – by requesting others to forward his packets, but avoiding to transmit others' packets.

Micro-payments are potential tools for fostering collaboration among selfish (rational) participants, and may be used to encourage collaborative routing of data and voice packets. However, while conceptually well suited to such a task, all proposed micro-payment schemes cannot be applied as such to the problem we consider here. One reason is that it is unlikely for a packet originator to know who – or even *how many* parties – are on the route of the packet. In contrast, traditional micro-payment schemes assume that the payer knows (at the very least) how many payments he is performing at any one time, but typically also whom he is paying – whether by identity, pseudonym or address. If the payer does not know whom he is paying, he could either attach several payment tokens (without any designated payee), or attach one token that can be deposited by several to him unknown parties. Either way there is a potential for abuse, and to avoid this, one needs to generate sufficient audit information to trace users who deposit more tokens than is appropriate. Previously proposed micro-payment schemes do not generate such audit trails.

To make things worse, we must not only consider the possible actions of individual cheaters, but also *collusions* of these. Here, a dishonest set of parties may do anything from routing a packet in a circular manner to claiming rewards for (or collecting payments on behalf of) parties not actually involved in the routing. A possible approach is to assume some degree of tamperproofness (as in [4]). While it can be argued that [4] and other related approaches rely on a similar form of tamperproofness as is successfully provided by GSM SIM cards [21,16], we mean that the latter provides "portability of identity" rather than security. This is because the SIM cards merely contain identifying information, and not accounting information, and an attacker cannot defraud others (whether other users or the operators) by modifying the functionality of his module. A better comparison in terms of adversarial setting may therefore be that of access to satellite entertainment. There, users may defraud the system (and routinely do) by using rogue modules. While satellite entertainment companies surely would prefer not having to rely on tamperproofness to assure correct behavior, they do not seem to have much choice. In contrast, and as we show, we do not have to rely on any form of tamperproofness to curb cheating – a careful protocol design suffices for the setting we consider.

Finally, we must consider the communication overhead (and the degree of interaction) necessitated by any solution, and make sure that this overhead is acceptable, even for the routing of single packets. This requirement is not normally placed on micro-payment schemes, where the primary constraint is often considered the computational requirement for performing – and receiving – a payment. We place emphasis both on the communication costs and the computation costs, noting that both of them translate into battery consumption.

Keeping this low by means of collaborative routing, of course, is the motivating force of this work, and the execution of our protocol must not depart from these goals.

Components and contributions. We avoid the use of all cryptographically heavy-weight operations, and make use of simple symmetric building blocks to achieve our goals. Thus, our contributions are not in the development of new cryptographic techniques, but rather, in addressing an important problem using the simplest possible building blocks. We propose an architecture that is suitable for the model, and put forward four different mechanisms that together constitute our protocol.

The *first* component is a technique for users to determine to whom a packet should be routed. Here, we allow each mobile device to have a preset threshold (potentially depending on its remaining battery life) corresponding to the size of the reward (or payment) they require to transport packets. Likewise, packet originators associate reward levels with packets according to the importance of having them transported. It must not be possible for cheaters to modify these reward levels, of course.

A *second* component is a technique allowing base stations to verify that all packets were accompanied by a valid payment, and drop those that were not. Given that we assume rational (as opposed to malicious) behavior of all participants, this rules out a denial of service attack in which a party causes transport of packets that will later be dropped by the base station due to their invalid payment fields. We argue that this is not a practical limitation, given that an attacker cannot completely drain anybody's batteries (even if constantly in their presence) since each mobile device has a threshold determining when they will collaborate. Moreover, there are easier ways of mounting denial of service attacks, such as simply jamming the communication channel.

A *third* component is a technique for aggregation of payments. Similar to the recent proposal by Micali and Rivest [17], this works by a probabilistic selection of payment tokens. As in [17], we allow this aggregation to be performed by the mobile devices (payees), for whom storage is a scarce resource. We also consider aggregation of payment information by the base stations as an additional cost-saving measure. To increase the granularity of payments, a user with a "winning ticket" would report the identities of his neighbors (along the packet's path) when filing a payment claim. Thus, not only the claimant is given a reward for transporting the packet, but his neighbors, too. While this allows for a reduction in storage requirements, its main use is within the fourth component:

The *fourth* component is an auditing process that allows the detection of cheating behavior. This is in the same spirit as the detection of reuse of sequence numbers in [17], but specific to our setting. In particular, our auditing techniques detect and trace dropping of packets, collusions of users filing payment claims, and attacks in which users give priority to the routing of packets carrying winning tickets. Our audit process takes advantage of already collected information from an array of different sources. First, it uses payment claims (winning tickets) from users. As mentioned above, these contain the identities of claimants' neighbors

(along the packet's path). Second, it uses packet transmission information from base stations. Third, it makes use of geographical location information collected by base stations – this is information about what users are in what cells at what time, and is already collected for other purposes.

Together, these mechanisms address the problem of how to foster collaboration among rational but selfish nodes[2] in a multi-hop cellular network. The two main contributions of our paper are the development of a suitable model and architecture; and an audit process suitable for detecting all important attacks without the need for the collection or maintenance of substantial amounts of data.

Outline. We begin by describing our technique at a very high level, and describe related work (section 2). We then turn to detailing our model and goals (section 3). Then, we describe our proposed protocol (section 4) and proposed accounting and auditing techniques (section 5).

2 Overview and Related Work

Multi-hop cellular networks. Although attractive at first sight, multi-hop cellular networks raise a number of problems. For example, in conventional cellular networks, base stations usually are in charge of channel allocation and of the synchronization and power control of mobile devices; to accomplish this task, they take advantage of their direct communication link with each and every mobile device currently visiting their cell. It is quite difficult to extend these operating principles to multi-hop cellular networks. A similar observation can be made in the framework of wireless LANs; for example, in an IEEE 802.11 network, a station can work either in infrastructure mode (namely, with one or several access points), or in ad hoc mode, but not in both.

Over the last years, several researchers have started to bring initial responses to the technical challenges of cellular multi-hop networks. The Soprano project [25] advocates self-organization of the physical, link and network layers. An analysis of the improvement of the throughput is provided in [12], while a routing protocol aiming at providing appropriate QoS is described in [13]. Connectivity of such networks is studied in [6] by means of percolation theory. The use of multi-hop networks is also envisioned in the third generation of cellular networks, where they are called "Opportunity Driven Multiple Access" (ODMA) [8].

Stimulating cooperation in Mobile Ad Hoc Networks. Several researchers have explored the problem of fostering cooperation (especially for packet forwarding) in mobile ad hoc networks. In [15] the authors consider the case in which some malicious nodes agree to forward packets but fail to do so. In order to cope with this problem, they propose two mechanisms: a *watchdog*, in charge of identifying the misbehaving nodes, and a *pathrater*, in charge of defining the best route

[2] In this paper, "node" and "mobile device" are synonymous.

circumventing these nodes. Unfortunately, this scheme has the drawback that it does not discourage misbehavior.

Another proposal [3] leverages on the reputation of a given user, based on the level of cooperation he has exhibited so far. In this scheme, users can retaliate against a selfish user by deying him service. It is important to note that this proposal is not restricted to packet forwarding but can encompass other mechanisms of the network. A drawback of this type of solution is that a set of colluding cheaters can give each other large quantities of positive feedback, while giving anybody critizing a member of the collusion negative feeback – both as a deterrrent and as a way to reduce the credibility of the feedback the honest user gave.

An even more recent contribution, called Sprite [27], takes a similar approach to what we do in our paper in that it considers an ad hoc network and assumes the presence of a backbone. On the other hand, it does not address the case of multi-hop cellular communications. While the contributions of Sprite are very nice in that they avoid assumptions on tamperproofness while still proving security statements for a stated model, there are potential drawbacks of their solution in terms of its overhead, security, and topology requirements. In particular, their scheme requires a fair amount of computation and storage, making it vulnerable to DoS attacks. Namely, Sprite requires the verification and storage of an RSA signature (or similar) for each packet. In contrast, we use faster verification functions (such as determining the Hamming distance between two strings) and only store verification strings for a fraction of all packets. Moreover, they do not consider attacks involving manipulation of routing tables, while we provide heuristic and statistic techniques to address this problem. Finally, they base their scheme on a reputation mechanism that will only be meaningful in rather dense networks. It is not clear that a typical network exhibits this property.

Among the related work, the already mentioned paper [4] is probably the closest to the present proposal in terms of the problems addressed and the main principles behind the solutions. While the trust model and the protocols of [4] are different from those we propose in that we do not rely on tamperproof hardware, the commonality is that of using micro-payments to foster collaboration in self-organizing networks.

A more general treatment of how to stimulate collaboration can be found in [18]. There, a theoretical framework for the design of algorithmic mechanisms is provided. Although it was developed in a different area, this approach could be applied to our problem, by considering that each node is an agent and that it has to accomplish specific tasks (such as packet forwarding).

Finally, for a general discussion of the security issues of mobile ad hoc networks and for a discussion on how key management can be made independent of any central authority, we refer the reader to [9].

We will now discuss the way we envision the use of payments in a multi-hop cellular network.

Our approach. Instead of using one payment token per payee (as is done in traditional micro-payment schemes [1,7,10,11,14,17,19,22,23]), we use one *per packet*,

letting all relaying nodes verify whether this token corresponds to a winning ticket for them. To avoid forged deposits, the packet originator needs a secret key to produce the token (not unlike other payment schemes.) To discourage colluders from collecting payments for each other, we require the intermediary's secret key (the same as is used to requesting service) to be used to verify whether a ticket wins. Thus, mutually suspicious colluders will not give each other their secret keys, as this allows the others to request service billed to the key owner.

Therefore, we propose a system in which all packet originators attach a payment token to each packet, and all intermediaries on the packet's path to a base station verify whether this token corresponds to a winning ticket. Winning tickets are reported to nearby base stations at regular intervals. The base stations, therefore, receive both reward claims (which are forwarded to some accounting center), and packets with payment tokens. After verifying the validity of the payment tokens, base stations send the packets (now without their corresponding payment tokens) to their desired destinations, over the backbone network. The base stations also send the payment tokens (or some fraction of these, and potentially in batches) to an accounting center. Packets with invalid tokens are dropped, as the transmission of these cannot be charged to anybody.

However, and again in contrast to previous micro-payment proposals, intermediaries are made to profit not only from *their own* winning tickets, but also from their neighbors' – we require all reward claims to be accompanied by the identities of the two neighboring parties on the packet's path. This has three direct benefits: First, the "neighbor reward" encourages the *transmission* of the packet, while the "personal reward" can be seen as a reward for *receiving* the packet – and for reporting this to the clearing house. Second, it increases the number of rewards per deposited ticket, which in turn means that fewer tickets need be deposited. Third, and more importantly, it allows for the compilation of packet forwarding statistics that can be used to detect inconsistent (read: cheating) behavior of intermediaries. By comparing the relative amounts of "neighbor rewards" and "personal rewards" on a per-node basis, the accounting center can detect various forms of abuse. In particular, this analysis will identify parties that routinely drop packets, and parties that refuse to handle packets without winning tickets. It will also detect various forms of collusion. As previously mentioned, we discourage users from performing "collaborative ticket checking" (when one party checks if a ticket is a winning ticket for one of his collaborators) by requiring that they know each other's keys for this to be possible. In addition, our auditing techniques allow for detection of such behavior, thereby providing two independent layers of protection. While the auditing techniques only detect *repeated* misbehavior (as opposed to the very occasional abuse), this is sufficient, as very few people are likely to alter their devices to make a few cents a month. On the other hand, the more aggressively somebody abuses the system, the faster they will be apprehended, and appropriately punished.

Relation to other payment schemes. Our aggregation principle is based on the idea of probabilistic payments[3], suggested by Rivest [22], and also related to work by Wheeler [24]. Therein, each payment can be thought of as a *lottery ticket*. Upon receiving it, the payee can determine whether it is a winning ticket or not, allowing him to erase tickets that were not winning, and request payments from the bank for those that were. While the bank only transfers funds for winning tickets, these are correspondingly larger, thereby causing the appropriate amount of funds to be transferred *on average*. In [22] and most other payment schemes that have been proposed, the bank is transferring funds between payers and payees in a "zero-sum" manner – meaning that for each payee that gets credited, the corresponding charge will be (or have been) levied upon the payer.

If the crediting is probabilistic (as it often is in micro-payments), this may mean that either payers or payees *perceive* a grand total charge (or credit) different than the number of payments suggests – at least for short periods of time. As pointed out by Micali and Rivest [17], this may result in difficulties getting such a scheme adopted by consumers. Accordingly, Micali and Rivest instead shift the temporary fluctuation to the bank, who debit and credit users according to the number of payments made resp. received. In particular, a payer is charged based on the number of payments he performs, while the payee is credited based on whether a payment contains a winning ticket or not. Micali and Rivest let each payment include a serial number that allows the bank to determine (from the winning and therefore deposited tickets) how many payments a given payer has performed. While it is possible for a payer to cheat the bank by performing several payments using the same serial number, this cannot be done *consistently* without the bank detecting it. This is so since the probability of one payment containing a winning ticket does not depend on the probability of another one containing a winning ticket – whether the payments use different serial numbers (as they should) or the same. Thus, while the audit mechanism does not necessarily detect a single instance of abuse, it *will* detect large-scale abuse.

Our proposal is somewhat similar to [17] in that payers (i.e., originators of packets) are charged per packet, and not per winning ticket, while users performing packet forwarding are paid per winning ticket. Therefore, the bank (or accounting center) in our scheme plays the same "averaging role" as the bank in [17]. The mechanisms for detecting abuse in our scheme are statistic, like those in [17]. That is, while it is possible for a payer to cheat the "bank" once, it is not possible in the long run. While there is only one payee per payer in traditional payment schemes, each node on a route in our scheme may win on a ticket associated with one specific packet. The sender pays a cost that – on average – covers the cost of routing, and of other network maintenance. The most straightforward approach of computing this charge is for the bank to compute the average uplink[4] cost (which depends on the reward level and the average number of hops) and include this per-packet charge in the general charges for transmitting

[3] In contrast, Jarecki and Odlyzko [10] perform probabilistic *audits*, while keeping the payments deterministic.

[4] The uplink is the link from the mobile device to the base station.

packets over the backbone. Such an approach is therefore a generalization of the averaging techniques proposed in [17].

While payees (i.e., nodes) in our scheme are not paid for each packet they handle, they are also not only paid corresponding to the winning tickets they collect: they are also paid each time a neighbor (along the packet's path) hands in a winning ticket. This provides a step in the direction of crediting payees *per transaction* by increasing the payment granularity, thus requiring fewer tickets to be handed in. It also provides an incentive for users to propagate packets carrying losing tickets (as they may be winning for the neighbor). Most importantly, though, this strategy supplies the back-end with a rich source of data from which it can detect protocol deviations.

As we have seen, payers are charged per packet, and not per winning ticket, while users performing packet forwarding are paid per winning ticket. Such asymetric payment schemes often allow a coalition of malicious users to make a net profit. As we will see in Section 5, our protocol is immune against this kind of abuse.

3 Model

User model. We assume the existence of three types of participants; *users*, *base stations*, and one or more *accounting center*. In addition, there may be multiple networks, each one of which is considered the *home network* for some users. We distinguish between base stations of the home network, and those of other networks, as will be explained below. We also assume that there is one accounting center per network.

Mobile devices usually have very limited storage and power resources. The base stations and the accounting center, on the other hand, correspond to powerful computers that are connected to each other by means of a high-bitrate backbone network.

Communication model. We assume the use of a network with a multiple-hop uplink, and a one-hop downlink, noting that this choice minimizes the global energy consumption of all mobile devices; we call a network of this kind "asymmetric multi-hop cellular network". In other words, as a packet travels from its originator to the closest base station, it is transmitted in multiple short hops, since this minimizes transmission costs. Here, the receiving base station may belong to the home network of the user, or (if the user is roaming) to another network, called the *foreign* network. Then, the packet is sent over the backbone from the base station receiving the packet, to the base station closest to the message recipient. (If the packet is multicast, there may be several such base stations, and corresponding receivers.) The closest base station, in turn, transmits it *directly* (i.e., in one hop) to the recipients – this does not require any involvement (or energy consumption) by any of the mobile devices in range. Note that the energy expenditures of the receiver are independent of the distance of the transmission: it is only the sender whose energy consumption depends on

this distance. This model is therefore different from the commonly used *symmetric* communication model in which cell phones and base stations communicate without intermediaries (i.e., where both uplink and downlink are one-hop).

This model is also different from the one usually considered for multi-hop cellular networks. Indeed, in the proposals published so far, both the uplink and the downlink connections are multi-hop. These properties are strongly influenced by the traditional approach of cellular networks (e.g., GSM) and wireless LANs (e.g., IEEE 802.11), in which all links are assumed to be bidirectional. This bidirectionality is considered to be very important, notably for radio resource allocation, power control, and synchronization.

The reason why we depart from this assumption is that a single-hop downlink can be highly beneficial. Indeed, as there is no need to relay downlink signals, the transmission power for the downlink is provided exclusively by the base station, sparing the batteries of the nodes which otherwise would have had to relay the packet. Moreover, this direct channel can be exploited to transmit synchronization signals from the base station to all mobile devices present in the cell. Finally, it makes the allocation of the radio resource on the downlink easier to implement. To the best of our knowledge, asymmetric multi-hop cellular networks have never been proposed in the literature; the study of their feasibility and of their potential merits and shortcomings is well beyond the ambitions of this paper.

Functional model. Users can be categorized as belonging to one or more of the following classes: *originators*; *recipients*; and *intermediaries*. An originator of a packet wishes to have this sent to one or more recipients of his choice. Intermediaries may act as routers, forwarding such packets towards the closest base station. Each such packet then gets transmitted through the backbone network to the base station(s) corresponding to the recipient(s); here they get broadcast by the base station in question and received by the desired recipient. Note again that a packet is only handled by intermediaries on its way *to* a base station, and not *from* a base station on its way to its recipient.

Trust model. Although in reality, very few consumers would attempt to modify the functionality of their devices, it is sufficient that a small fraction would abuse the protocol in order for its commercial usefulness to be endangered. Reflecting this, we make the pessimistic assumption that the devices can be straightforwardly modified by their owners, corresponding to modelling the user as a software module run on a multi-purpose computer, with an appropriate communication module. Users are not trusted to act according to the protocol, but rather, may deviate from this in any arbitrary way. However, it is assumed that the users act *rationally*, i.e., that they only deviate from the protocol when they can benefit from doing so. In particular, users could collude in an arbitrary fashion, and could use a strategy that is a function of data they receive by means of the network. Users trust base stations of their home network not to disclose their secret keys; no such trust has to be placed in base stations outside their home network. All base stations are trusted to correctly transmit packets,

and to forward billing and auditing information to the accounting center of the user's home network, according to the protocol. The accounting center, in turn, is trusted to correctly perform billing and auditing. These are reasonable assumptions for a network that is well guarded against compromise; it is also a reasonable assumption in a network constituting of a small number of principals that audit each other's activities, both by cryptographic/statistical means, and by traditional means.

Goals. The end goal of our protocol is to maximize battery life by minimizing the required transmission signal strength of mobile devices, with the added benefit of increasing the available bandwidth by reducing signal strength. In order to reach this goal, given the selfish nature of users, we propose a set of mechanisms for encouraging collaboration and detecting (and punishing) cheating. In particular, these mechanisms are designed to address several types of abuse, as described hereafter.

Abuse. A naïve solution to the problem may simply provide users with a strategy that maximizes the common good by requiring individual users to collaborate by forwarding other users' packets. However, users – being selfish – may deviate from this proposed protocol. In order to reward altruism, our protocol aims to detect collaboration, allowing this to be rewarded – whether in monetary terms or in terms of improved service levels. Furthermore, our protocol has mechanisms for detection of various forms of cheating. In particular, we prevent or detect the following types of abuse, whether these strategies are used in a "pure-bred" form, or in combination with each other:

- **Selective acceptance.** A cheating strategy in which a user agrees to receive (with the intent to re-transmit) packets with winning tickets, but not packets without winning tickets. (A variation of the attack is when a first user sends a packet to a friend to route, given that the packet is likely to contain a winning ticket for the friend.)
- **Packet dropping.** When a user agrees to receive packets, but does not re-transmit them – whether he claims credit for winning tickets or not.
- **Ticket sniffing.** When a user claims credit for packets he intercepted, but neither agreed to re-transmit nor actually re-transmitted. In a severe version of this attach, colluding users along a fake path submit claims as if they routed the packet.
- **Crediting a friend.** When a user with a winning ticket claims to have received the packet from (or have sent it to) a party different from that which he in actuality did receive it from (resp. sent it to.)
- **Greedy ticket collection.** This is a collection of cheating strategies aimed towards allowing users to claim credits in excess of what the protocol specifies, by collecting and sharing tickets with colluders. Three special cases of this general attack are (1) when one user collects tickets for a friend, knowing that these are likely to be winning tickets for the friend; (2) when sets of users collect and pool tickets, allowing each other to sift through a larger

pool than they routed; and (3) when a user obtains two or more identities, evaluating tickets with all of these to increase the chances of winning.
- **Tampering with claims.** An attack in which a cheater modifies or drops the reward claim filed by somebody else – when routed via the cheater – with the goal of either increasing his profits or removing harmful auditing information.
- **Reward level tampering.** An attack in which a packet carries an "exaggerated" reward level promise during some portion of its route, but where the reward level indicator is reduced before it is transmitted to the base station.

Note, however, that a plain refusal to collaborate is not abuse, as long as the refusal is independent of whether a packet carries a winning ticket or not. Users may choose not to route other users' packets if their resources or policies do not permit them to do so.

Moreover, note that we do not address "circular routing" as a possible attack, given that the rewards will be deterministic given a particular ticket, and therefore, such routing does not behoove an attacker. Neither do we consider the milder form of abuse where a set of users route a message along an unnecessarily long path within a particular neighborhood, in order to allow all of them to (justifiably) claim credit for having handled the packet – this assumption is reasonable if there is enough "real traffic" to route, and the reward structure is set appropriately.

4 Protocol

Setup. As a user u registers to be allowed access to the home network, he is assigned an identity id_u and a symmetric key K_u. This pair is stored by the user and by the user's home network. As is common, users offer their service provider some form of security, normally implemented by means of a contract or deposit.

Rewards. Originators may indicate one of several reward levels; the ultimate (billing) cost for these levels will be specified by his service agreement. The reward level L is an integer within a pre-specified interval $[0 \ldots max_L]$. Intermediaries are rewarded accordingly: if transmitting a packet associated with a higher reward level, their expected reward will be greater (with reimbursement levels specified by their service contract). Increasing the reward level allows users with particularly low battery resources to obtain service in a neighborhood populated by other users with low battery resources.

Connectivity graph. We assume that each user u keeps a list[5] λ_u of triples (u_i, d_i, L_i), where u_i is the (unique) identity of a neighbor with a path of length

[5] We do not address how the routing table is built, noting that any standard method, whether proactive or reactive, may be employed. In addition to standard information, the users also exchange information about their reward thresholds L_i.

d_i hops to the closest base station. Furthermore, L_i is user u_i's corresponding threshold for forwarding packets. (Thus, an entry (u_i, d_i, L_i) in λ_u means that user u_i will forward all packets whose reward level is equal to or greater than L_i, and that the length of the path from u_i to the base station is d_i.) We assume that λ_u is sorted in terms of increasing values of d_i, and that all entries with the same distance d_i sorted in terms of increasing values of L_i.

Packet origination. The *originator* u_o of a packet p selects a reward level $L \in [0 \ldots max_L]$, and computes a MAC $\mu = MAC_{K_{u_o}}(p, L)$. He then assembles the tuple (L, p, u_o, μ) and transmits this according to the transmission protocol below.

Packet transmission. Let u be a user (whether originator or intermediary) who wishes to transmit a packet associated with a tuple $P = (L, p, u_o, \mu)$. In order to transmit P, user u performs the following protocol:

1. If the base station can be reached in a single hop, then u is allowed to send the packet directly to it; otherwise he goes to step 2.
2. u selects the first (hitherto unselected) entry (u_i, d_i, L_i) from λ_u for which $L_i \leq L$.
3. u sends a *forward request* to u_i. This contains the reward level L and possibly further information about the packet p, such as its size[6].
4. u waits for an acknowledgement from u_i for some pre-set time period δ. If u receives the acknowledgement, then he sends P to u_i. Otherwise, if no acknowledgement arrives, he increases i by one. If $i > |\lambda_u|$ then he drops the packet; otherwise, he goes to step 2.
5. If u is not the originator of the packet, he performs the reward recording protocol below.

Packet acceptance. Let u' be a user receiving a forward request from u with reward level L. If L is less than his threshold, then he does not accept the request; otherwise, he accepts it by sending an acknowledgement to u and awaits the transmission of the packet.

Network processing. When a packet $P = (L, p, u_o, \mu)$ is received by a base station in the originator's home network, the base station looks up the secret key K_{u_o} of the originator u_o, and verifies that $\mu = MAC_{K_{u_o}}(p, L)$, dropping the packet if this does not hold.

If the packet is received by a base station that belongs to a foreign network, this base station cannot perform the verification (as it does not have access to the originator's secret key), and so, forwards the packet P to a register in the originator's home network. This register, then, looks up the originator's secret key, performs the verification, and drops the packet if the verification fails[7].

[6] Most protocols support several packet sizes.

[7] Similarly to the technique adopted in most 2G and 3G cellular networks, the detour of each and every packet via the home network can be avoided, by letting the foreign network perform the described verifications; this can be done without revealing the secret key to the foreign network.

If the verification of the MAC succeeds, the base station (resp. home network register) transmits the packet portion p to the base station associated[8] with the desired recipient (as indicated in p). The base station associated with the desired recipient broadcasts p to the latter.

The first base station records a fraction ϵ_μ of all triples (μ, L, u), where u is the identity of the user it received the packet from. It also keeps a count cnt_{u_o} of the number of packets it transmits for u_o. Periodically, base stations send such recorded auditing information to an accounting center, along with geographical information consisting of statistics of what users were in what cell at what time (not all such information is sent, but some portion.)

Reward recording. After user u has forwarded a tuple $P = (L, p, u_o, \mu)$, he verifies whether $f(\mu, K_u) = 1$ for some function f (the choice of which is discussed below). If this relationship holds, it means that the considered ticket is winning; he then records (u_1, u_2, μ, L), where u_1 is the identity of the user he received the associated packet from, and u_2 is the identity of the user (or base station) he forwarded it to. We let M denote the list of recorded reward triples.

Reward claim. If a user u is adjacent to a base station (i.e., the distance to the base station is 1), then he transmits a claim (u, M, m) to the base station, where $m = MAC_{K_u}(hash(M))$. Thus, the reward claim M is authenticated using the same key K_u as the user employs when originating a packet, or verifying whether a packet contains a winning ticket.

Similarly, if user u originates a packet P or is running out of storage space for claims, then he transmits the claim to the closest base station by means of the packet origination protocol, and using the base station as the packet recipient. The portion M may be encrypted using a stream cipher and using a secret key shared by user u and either the base stations or the accounting center – in the latter case, the MAC m would be computed on the ciphertext of M.

When a base station receives a claim, it verifies the correctness of the MAC m with respect to the user u and the claim M (or the ciphertext, as explained above). If this is not correct, then he ignores the claim; otherwise, he records the claim and computes an acknowledgement ack to it as $ack = MAC_{K_u}(m)$, where K_u is the key he shares with the user (claimant) u. ack is transmitted to u, who upon receipt verifies the acknowledgement and erases M if correct. Within a time Δ, each base station forwards all recorded claims to an accounting center, and then erases the list.

[8] Standard techniques can be used to determine in what cells packet recipients are located. In particular, one may require users to announce their location to base stations at regular intervals, or to announce changes of location – inferred by these by the changing identity of the closest base station. While this "announcement" is currently performed by direct communication from mobile device to base station, our multi-hop technique can obviously be used instead. Users may piggyback reward claims with such location announcements.

Ticket evaluations. As mentioned above, all tickets μ are evaluated with respect to the secret[9] key K_u of the user u in question, and with respect to some public function f that results in a uniform distribution of winning tickets. One can choose f as a one-way function, such as a hash function, and let a winning ticket be one that hashes to a value with a certain pattern (e.g., any string that starts with ten zeroes.) However, since the evaluation of f has to be performed once for each packet the user u handles (except for those packets originating with u, of course), it is important that f is lightweight, and preferrably more light-weight than hash functions are.

A promising possibility is to let $f(\mu, K_u) = 1$ iff the Hamming distance between μ and K_u is less than or equal to some threshold h. Thus, assuming that $|\mu| = |K_u|$, and given a particular reporting threshold h, the probability of μ being a winning ticket is

$$\frac{1}{2^\ell} \sum_{i=0}^{h} \binom{\ell}{i}$$

where $\ell = |\mu| = |K_u|$. Note that it is possible to assign different rewards to different Hamming weights in the range, making it possible for a user to keep only the "highest rewards" in case he runs out of memory and needs to purge some portion of the rewards. However, for simplicity, we assume that all reward claims have the same value.

However, we note that if f is not a one-way function (as in the case above) then it may be possible for an attacker to derive the user's secret key K_u by observing what tickets are filed. Therefore, if such a function is used, it is important that all claims are encrypted during transmission, in which case only the *number* of claims (as opposed to the form of these) would be revealed to an attacker.

We note also that f must be chosen in a way that the distribution of winning tickets is uniform.

On the probability of winning. The efficiency of our protocol relies on the probability of a ticket to win to be small enough for the claim process not to dominate the protocol, whether in terms of storage or communication. At the same time, we need the probability to be large enough that the reimbursement process relies on a large number of claims, which in turn makes auditing possible by providing a sufficiently large data set. Therefore, one needs to carefully balance these problems against each other when selecting the appropriate reward function. Rather than a security issue, this corresponds to a risk management issue and a useability issue.

[9] It is important that all of (or close to all of) K_u is needed to evaluate f successfully – or users would be able to verify reward claims on behalf of each other, without having to trust each other with their secret keys.

5 Accounting and Auditing

The accounting center receives both user claims and partial transmission transcripts – both forwarded by base stations. These are processed as follows:

Accounting. The accounting center periodically verifies all received user claims with respect to all recorded reward tuples it has received from base stations. All originators whose identity, u_o, has been recorded by a base station are charged a usage fee according to their service contract. Moreover, the accounting center credits all parties[10] whose identity figures (whether as a claimant or neighbor thereof) in an accepted reward claim. It is a policy issue how to set the rewards for neighbors to claimants, i.e., whether to let these depend on the reward level of the packet as well, and how large a neighbor reward would be in comparison to a claimant reward.

Here, a reward claim is said to be *accepted* if it is *correct* (i.e., if $f(\mu, K_u) = 1$) and a base station has reported the packet associated to the ticket μ as having been transmitted. Note that the accounting center may credit claimants and neighbors thereof according to any policy, and, in particular, the amounts may differ between claimants and neighbors; we do not dwell on these intricacies in this extended abstract.

Simplified auditing. Assume for a moment that the probability of a ticket to win is 1, and that all of these claims get reported by the users and passed on to the auditing center. Assume further that all MAC headers are stored by the base stations, and forwarded to the auditing center. We can now see that the auditing center will know the origination point of each packet (from the identity and MAC of the packet), and the identity of the base station receiving it. It will also know the identity of the user transmitting it to the base station (since this is recorded by the latter). From the claim of this user, it will know the identity of the user one step earlier in the forwarding chain, and so on. This will take us all the way back to the identity of the user who received the packet from the originator, who in turn will report whom he received it from (i.e., the originator.) If any user other than those already accounted for in the above claims a reward, this will be identified as an attempt to cheating. The auditing process for the probabilistic setting is analogous to the analysis of the simplified setting in that it approximates the latter by means of statistical methods.

Auditing. In the following, we will assume that $\epsilon_\mu = 1$, i.e., each base station stores the MAC header of each packet. The more general case in which different base stations store different fractions can be dealt with similarly: instead of merely counting occurrences, one would then test various hypotheses using standard statistical methods. It is worth noting that the probability of a ticket being a winning ticket is a function of three quantities: the message; the secret

[10] There are two exceptions to this rule: Neither packet originators nor base stations obtain rewards.

key of the originator; and the secret key of the intermediary (i.e., the party verifying whether the ticket is winning.) Since the secret keys of users are selected uniformly at random, the distribution of winning tickets is uniformly distributed over all messages.

Common for many of the detection mechanisms is the observation that since the probability for a ticket to win is independent of the identity of the user, each user should figure as the claimant with approximately the same[11] frequency as he figures as either the *sending neighbor* or *receiving neighbor* of a claimant. While one cannot simply compare the number of occurrences of these events, one can check the hypothesis that they are all generated from a source with the same event probability. As will become evident, many of the attacks we consider leave very similar-looking evidence, which may make it difficult to establish with certainty what the attack was. However, one can easily establish the presence of one of these attacks using standard statistical methods, and given sufficient material.

- **Selective acceptance.** Selective acceptance is epitomized by a user figuring as a *claimant* with a significantly higher frequency than as a *sending neighbor*.
- **Packet dropping.** A user is suspected of packet dropping if he has a higher *claimant* frequency than *sending neighbor* frequency for packets that were not reported as received by any base station.
- **Ticket sniffing.** A user is suspected of ticket sniffing if he has a higher claimant frequency than *sending neighbor* or *receiving neighbor* frequency, and there are incidents when both he and a neighbor files a claim for one and the same ticket, but do not list each other as the corresponding neighbors. If an entire fake path of reward claims has been created, the auditing center can distinguish between this and a real path (with some probability) given that the receiving base station will record the identity of the user from whom the packet was received.
- **Crediting a friend.** An indication of this attack is that the *receiving neighbor* of a given claim was reported by the base station to have been located in a distant cell[12] at the time the packet was received by the base station. Another indication is if a first user reports a second party to be the receiving neighbor, while another (also claiming a reward for the same packet) claims to have received the packet from the first party. While it is difficult to determine from one occurrence whether the first or the third party filed an incorrect claim, repeated occurrences will allow this to be established.

[11] We note that this is true independently of what the "collaboration thresholds" of the different parties on the route are. This is so since we consider the frequencies along a path of senders where all have agreed to collaborate – their thresholds are therefore irrelevant!

[12] All cellular devices report to the closest base station when they move from one cell to another. Similarly, when a device is turned on, it reports to the closest base station. If a device is moved while turned off, we consider it to still remain in the cell where it last was heard from.

- **Greedy ticket collection.** This has the same symptoms as the above mentioned attack. In addition, transmission paths – counted in number of claims per packet – that are longer than usual (for the given cell) are indicative of this attack. Similarly, abnormally high packet transmission rates per time unit by some user indicates that greedy ticket collection has taken place. Unusually large numbers of reward claims per time period therefore suggests that this has taken place. (We note that the transmission rates must be placed in the context of what type of hardware is used. The hardware type is likely to be known by the service provider, so this does not cause any problem.)
  The greedy ticket collection attack is likely to be the hardest attack to detect; especially if users scan for tickets of packets sent within the same cell as they resided, and if the users take pains to make the reported neighbors consistent with each other. However, should one party be found guilty of this attack, this is likely evidence that its common neighbors are, too.
- **Tampering with claims.** This attack is prevented by use of authentication techniques; the use of auditing tools does therefore not relate to the securing against it.
- **Reward level tampering.** If claimants indicate higher reward thresholds than that used for a given packet, this is an indication that the originator and some colluder close to the base station may perform this attack. Repeated evidence from different claimants, all pointing towards one and the same originator, provides strong evidence of the attack, in turn.

As for credit card fraud, use patterns can be employed to guard against attacks; the above description is meant only as evidence that the collected audit information is sufficient to detect and trace misbehavior. We are aware of further techniques to do so, and believe that there are further techniques we are not aware of. In fact, this problem is quite similar to intrusion detection, which has been studied for most existing and envisioned networks, including mobile ad hoc networks [26].

6 Conclusion

We have described an architecture for fostering collaboration between selfish nodes of multi-hop cellular networks, and have provided mechanisms to encourage honest behavior and to discourage dishonest behavior. To the best of our knowledge, no single paper was published so far on this issue.

Our security model is not formal: Instead, we list a set of potential abuses along with associated detection mechanisms. Thus, we propose to deal with fraud in a similar manner to how telecommunications companies and credit card companies do. A less heuristic approach would be a great step forward; however, this is a difficult task. Part of the reason for this is that not all packet forwarding information gets reported to the auditor (as not all tickets are winning), and that honest users may lose connectivity at any time. However, even if that were

not the case, a formal approach appears to be non-trivial. We hope that our contribution can be a first step in the direction of a formal treatment of the problem.

In terms of future work, we intend to work on this formalization. In addition, we will relax the assumption that all packets have to go through the backbone, by combining the proposed solution with an approach related to pure ad hoc networks, such as the one proposed in [4]. Moreover, we will explore the symmetric case of multi-hop cellular networks, and estimate the performance of the proposed solution and propose appropriate optimizations whenever necessary. Finally, we will consider session-based (as opposed to packet-based) solutions; a result representing a first step in that direction will appear shortly [2].

Acknowledgements. We wish to thank Philippe Golle, Ari Juels and Ron Rivest for helpful discussions and feedback.

References

1. R. Anderson, H. Manifavas, C. Sutherland, "A Practical Electronic Cash System," In proceedings Fourth Cambridge Workshop on Security Protocols, 1996.
2. N. Ben Salem, L. Buttyán, J.-P. Hubaux, M. Jakobsson, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-Hop Cellular Networks," to appear in MobiHoc '03.
3. S. Buchegger, J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes: Fairness in Dynamic Ad-hoc NeTworks)," Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, June 2002 (MobiHoc 2002)
4. L. Buttyan, J. P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks, October 2003, Vol. 8 No. 5
5. M. Cagalj, J. P. Hubaux, C. Enz, "Minimum-Energy Broadcast in All-Wireless Networks: NP-completeness and Distribution Issues," Proceedings of the Eighth ACM International Conference on Mobile Networking and Computing, Atlanta, September 2002 (Mobicom 2002)
6. O. Dousse, P. Thiran, M. Hasler, "Connectivity in Ad Hoc and Hybrid Networks", 21st Annual Joint Conference of the IEEE Computer and Communications Societies, New York, 2002 (Infocom 2002)
7. R. Hauser, M. Steiner, M. Waidner, "Micro-Payments Based on iKP," Technical Report 2791 (# 89269), June 1996.
8. H. Holma, A. Toskala, "WCDMA for UMTS", Wiley 2000
9. J.-P. Hubaux, L. Buttyan, S. Capkun, "The Quest for Security of Mobile Ad Hoc Networks," Proceedings of the Second ACM International Symposium on Mobile Ad Hoc Networking and Computing, Long Beach, October 2001 (MobiHoc 2001)
10. S. Jarecki, A. Odlyzko, "An Efficient Micropayment System Based on Probabilistic Polling," Financial Cryptography '97, pp. 173–191
11. C. Jutla, M. Yung, "PayTree: "amortized-signature" for flexible MicroPayments," Proceedings of Second USENIX Workshop in Electronic Commerce, pp. 213–221, 1996.

12. Y.-D. Lin, Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications", 19th Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, 2000 (Infocom 2000)
13. C. R. Lin, "On-Demand QoS Routing in Multihop Mobile Networks", 20th Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage, 2001 (Infocom 2001)
14. M. Manasse, "Millicent (electronic microcommerce)," 1995. `www.research.digital.com/SRC/personal/Mark_Manasse/uncommon/ucom.html`.
15. S. Marti, Th. Giuli, K. Lai, M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proceedings of the Sixth ACM International Conference on Mobile Networking and Computing, Boston, August 2000 (Mobicom 2000)
16. A. Mehrotra , L. Golding, "Mobility and security management in the GSM system and some proposed future improvements," Proceedings of the IEEE, vol. 86, no. 7, July 1998, pp. 1480–1496.
17. S. Micali, R. Rivest, "Micropayments Revisited," CT-RSA 2002, pp. 149-163
18. N. Nisan, A. Ronen, "Algorithmic Mechanism Design," Proceedings of the 31st ACM Symposium on Theory of Computing, 1999, pp. 129–140
19. T. Pedersen, "Electronic Payments of Small Amounts," Technical Report DAIMI PB-495, Aarhus University, Computer Science Department, Aarhus, Denmark, August 1995.
20. C. Perkins, "Ad Hoc Networking," Addison Wesley, 2001.
21. M. Rahnema, "Overview of the GSM system and protocol architecture," IEEE Communications Magazine, vol. 31, no. 4, April 1993, pp. 92–100.
22. R. Rivest, "Electronic Lottery Tickets as Micropayments," Financial Cryptography '97, pp. 307–314
23. R. Rivest, A. Shamir, "Payword and MicroMint – two simple micropayment schemes," Proceedings of 1996 International Workshop on Security Protocols, pp. 69–87, 1996.
24. D. Wheeler, "Transactions Using Bets," In proceedings Fourth Cambridge Workshop on Security Protocols, pp. 89–92, 1996.
25. A. Zadeh, B. Jabbari, R. Pickholtz, B. Vojcic, "Self-Organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)", IEEE Communications Magazine, June 2002
26. Y. Zhang, W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," Proceedings of the Sixth ACM International Conference on Mobile Networking and Computing, Boston, August 2000 (Mobicom 2000).
27. S. Zhong, Y. R. Yang, J. Chen "Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad Hoc Networks", Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, July 2002

On the Anonymity of Fair Offline E-cash Systems

Matthieu Gaud and Jacques Traoré

France Télécom R&D
42 rue des Coutures, BP 6243
14066 Caen Cedex 4, France.
{matthieu.gaud, jacques.traore}@francetelecom.com

Abstract. Fair off-line electronic cash (FOLC) schemes [5,29] have been introduced for preventing misuse of anonymous payment systems by criminals. In these schemes, the anonymity of suspicious transactions can be revoked by a trusted authority.
One of the most efficient FOLC system has been proposed by de Solages and Traoré [13] at Financial Cryptography'98. Unfortunately, in their scheme, the security for legitimate users (i.e., anonymity) is not clearly established (i.e., based on a standard assumption).
At Asiacrypt'98, Frankel, Tsiounis and Yung [17] improved the security of [13] by proposing a fair cash scheme for which they prove anonymity under the Decision Diffie-Hellman (DDH) assumption.
In this paper, we show that Frankel et al. failed to prove that their scheme satisfies the anonymity property. We focus here on this security problem and investigate the relationships between different notions of *indistinguishability* in the context of fair electronic cash. As a result, we prove under the DDH assumption, that a straightforward variant of [13], which is more simple and efficient than [17], is secure for users. This proof relies on the subsequent result of Handschuh, Tsiounis and Yung [19] showing equivalences between *general* decision and matching problems. Our proof is somewhat generic and can be used to prove that [17] is secure as well.

1 Introduction

Many anonymous electronic cash systems have been proposed in the recent years. In these systems, there is no mechanism for the bank, the merchants or any other party to identify the users involved in a transaction. If desirable from a user's point of view, this unconditional anonymity could however be misused for illegal purposes, such as money laundering or perfect blackmailing [32].

Fair electronic cash systems have been suggested independently by [5] and [29] as a solution to prevent such fraudulent activities. The main feature of these systems is the existence of a trusted authority (trustee) that can revoke, under specific circumstances, the anonymity of the coins. Unfortunately the first fair cash schemes [5,8,20] required the participation of the trustee in the opening of an account or even in the withdrawals of coins, which is undesirable in practice.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 34–50, 2003.
© Springer-Verlag Berlin Heidelberg 2003

Camenisch, Maurer and Stadler [6] and independently Frankel, Tsiounis and Yung [16] proposed fair e-cash schemes with an *off-line* (passive) authority: the participation of the trustee is only required in the set-up of the system and for anonymity revocation. The main difference between these two systems ([6] and [16]) is related to the tracing mechanisms: in [16], the trustee directly finds the identity of the owner of a specific coin (*owner tracing*), whereas in [6] the trusted authority performs a search in a large database (the database of withdrawal coins).[1] In [7], Camenisch et al. extend their scheme [6] to the setting of wallets with observers while [12] simplified the protocol of [16] using faster coin tracing techniques.

At Financial Cryptography'98, de Solages and Traoré [13] presented more efficient fair cash schemes than those of [7] and [12]. However, the security for legitimate users (i.e., anonymity) is not formally proven and seems to rely on a stronger variant of the DDH assumption.

At Asiacrypt'98, Frankel, Tsiounis and Yung improved the security of [13] by proposing a fair cash scheme for which they prove anonymity under the DDH assumption. Their proof is based on the equivalence of the semantic security of ElGamal encryption [14] and the DDH assumption. All these schemes [6,7,16,13, 17] are constructed from Brands' anonymous payment system [4]. A simple and efficient solution for the Digicash^{TM} system has been proposed by Juels [22].[2]

In this paper, we show that the proof in [17] is incorrect. Our aim, similar to the one of [17], is then to propose a simpler scheme, with the minimal number of additions required for provable anonymity. Our proposition is a straightforward variant of [13]. We prove anonymity based on the DDH assumption. We show in particular the relationships between two different notions of *indistinguishability* in the context of fair electronic cash. As a result, we are able to prove that [17] is secure as well.

Several researchers have criticized the *trustee based tracing* model [5,6,7,16, 13,17]. Among the criticisms is the fact that the revocation ability could be misused by the trustee himself (without anyone being able to detect such illegal tracings).[3]

Pfitzmann and Sadeghi [26] have thus introduced the concept of *self-escrowed cash* to solve the problem of user blackmailing. Instead of a trustee, it is the user himself who is able to trace his own coins in case of blackmailing.[4] Thus the risk of misuse of the revocation ability is eliminated.

Kügler and Vogt [23] have introduced the concept of *auditable tracing*: the users

[1] In [1], Abe and Ohkubo describe a concrete vulnerability of such schemes (which only perform owner tracing against the withdrawal database). See [1] for more details. To prevent such attack, the parameter α used in the withdrawal protocol of [6] has to be jointly computed by the bank and the user, which greatly impacts the efficiency of their scheme.

[2] See also [30] for a different approach based on group signatures.

[3] An "orwellian" like trustee.

[4] In [26], a concrete instantiation of self-escrowing was implemented based on the system from [16]. In [24], Meier shows how to transform the FOLC systems of [6,13, 17] in ones with self-escrowing.

can later audit their payments and detect whether their spent coins have been traced or not. Illegal tracings by the bank (without the permission of a judge) could then be prosecuted.

Our scheme supports, with minor and straightforward modifications, self-escrowing as well as auditable tracing.

Organization of the paper: The variant of [13] is decribed in section 2. In section 3, we explain why the proof of anonymity in [17] is wrong.[5] In section 4, we prove that the variant is secure for legitimate users. In Appendix, we recall the background of the key techniques used in [13].

2 The Modified Fair Cash Scheme

In the simplified model of fair electronic cash that we use, four types of parties are involved: a bank $\mathcal{B}$, a trusted authority $\mathcal{T}$, shops $\mathcal{S}$ and users $\mathcal{U}$. A fair e-cash scheme consists of five basic protocols, three of which are the same as in anonymous e-cash, namely a withdrawal protocol with which $\mathcal{U}$ withdraws electronic coins from $\mathcal{B}$, a payment protocol with which $\mathcal{U}$ pays $\mathcal{S}$ with the coins he has withdrawn, and a deposit protocol with which $\mathcal{S}$ deposits the coins to $\mathcal{B}$. The two additional protocols are conducted between $\mathcal{B}$ and $\mathcal{T}$, namely *owner tracing* and *coin tracing* protocols. They work as follows:

- *coin tracing protocol*: the bank provides the trusted authority with the view of a withdrawal protocol and asks for the information that allows it to identify the corresponding coin in the deposit phase.
- *owner tracing protocol*: the bank provides the trusted authority with the view of a (suspect) payment and asks for the identity of the withdrawer of the coins used in this (suspect) payment.

In the following, we describe a FOLC scheme based on [13] which is provably anonymous. We will adopt the notation and techniques used in [17] and [13]. We refer interested readers to these papers for more details on these techniques. (See also Appendix for a description of some of these basic key techniques).

2.1 The Setup of the System

Notation and Definitions. Throughout the paper we will use the following notation: the symbol $\|$ will denote the concatenation of two strings. If x is an integer, $|x|$ will denote the binary size (length) of x. The symbol ϵ will denote the empty string. The notation "$x \in_R E$" means that x is chosen uniformly at random from the set E. The notation '$x \stackrel{?}{=} y$', used in a protocol, means that the party must check whether x is equal to y. It is assumed that if the verification fails, the protocol stops.

Let $\mathbb{Z}_n$ denote the residue class ring modulo n and $\mathbb{Z}_n^*$ the multiplicative group of invertible elements in $\mathbb{Z}_n$. We denote by G_q a group of large prime

[5] We limit our analysis to the protocol of section 5 of [17] (the simplified FOLC).

order q, such that computing discrete logarithms in this group is infeasible. For $g, h \in G_q$, $g \neq 1$, we let $\log_g h$ denote the discrete logarithm of h to the base g, which is equal to the unique $x \in \mathbb{Z}_q$ satisfying $h = g^x$. A common construction of G_q is to take the unique subgroup of order q in $\mathbb{Z}_p^*$, where p is a large prime such that $q \mid p-1$.
For any integer m, for any element h and distinct generators $g_1, g_2, ..., g_m$ in G, we say that $(x_1, x_2, ..., x_m) \in \mathbb{Z}_q^m$ is a *representation* of h with respect to the base $(g_1, g_2, ..., g_m)$ if $h = \prod_{i=1}^m g_i^{x_i}$. The value (coordinate) $x_i, i \in \{1, ..., m\}$ is called the discrete logarithm of h with respect to g_i in the representation.
Finally, we denote by $\mathcal{H}$ a collision-resistant hash function, and we use $\mathcal{H}(a, b)$ (or $\mathcal{H}(a\|b)$) to denote the image under $\mathcal{H}$ of the concatenation of the strings a and b. For convenience, we assume that the range of $\mathcal{H}$ is equal to $\mathbb{Z}_q$.
Let $SK[(\alpha, \beta, ...) : Predicates](m)$ be the *signature of knowledge* on the message m proving that the signer knows $(\alpha, \beta, ...)$ satisfying the predicate *Predicates*. In this notation, Greek letters will denote the secret knowledge and the others letters will denote public parameters between the signer and the verifier (see Appendix for more details).

Bank's Setup Protocol: (performed once by $\mathcal{B}$) Primes p and q are chosen such that $|p-1| = \delta + n$ for a specified constant δ, and $p = \gamma q + 1$ for a specified integer γ. Then a unique subgroup G_q of prime order q of the multiplicative group $\mathbb{Z}_p^*$ and generators g_1, g_2, g_3, g_4 of G_q are defined.[6] Secret key $x_{\mathcal{B}} \in_R \mathbb{Z}_q^*$ is created.[7] $\mathcal{B}$ also determines a collision-free hash function $\mathcal{H}$ that maps $\{0,1\}^*$ to $\mathbb{Z}_q$. $\mathcal{B}$ publishes p, q, g_1, g_2, g_3, g_4, $\mathcal{H}$ and its public keys $h_1 = g_1^{x_{\mathcal{B}}}$, $h_2 = g_2^{x_{\mathcal{B}}}$, $h_4 = g_4^{x_{\mathcal{B}}}$.

Trustee's Setup Protocol: (performed once by $\mathcal{T}$) $\mathcal{T}$ chooses a secret value $x_{\mathcal{T}} \in_R \mathbb{Z}_q^*$ and publishes its public keys $f_2 = g_2^{x_{\mathcal{T}}}$ and $f_3 = g_3^{x_{\mathcal{T}}}$. $\mathcal{B}$ then publishes $F = f_3^{x_{\mathcal{B}}}$.[8]

Opening an Account: (performed for each new user $\mathcal{U}$) At account opening, $\mathcal{U}$ chooses a secret value $u \in \mathbb{Z}_q^*$ and computes his public identity or "account number" as $I = g_1^u$. Then, $\mathcal{U}$ must prove to $\mathcal{B}$ (using the Schnorr identification scheme [28]) that he knows the discrete logarithm of I to the base g_1.

[6] It is assumed that no representation of either of these elements with respect to the others is known.
[7] For the sake of simplicity, we assume that there is only one coin denomination in the system (extension to multiple denominations is easy).
[8] We assume that there is only one trusted authority $\mathcal{T}$ (extension to several trustees is easy).

2.2 The Withdrawal Protocol

The withdrawal protocol consists of two phases: the coin tracing phase and the coin withdrawal phase. During the coin tracing phase, the user gives the bank the information that will enable the trusted authority to recognize the withdrawn coin after it has been spent. In the coin withdrawal phase, the user obtains a blind signature on his coin. Here we will use the blind signature protocol defined in [13] (called *BlindSig* in their paper). The signature scheme underlying *BlindSig* is the one presented in [11]. Let us briefly recall this scheme.

The Chaum-Pedersen Signature Scheme

The Parameters. The following parameters are generated by the signer (the bank here). Primes p and q are chosen such that $|p-1| = \delta + n$ for a specified constant δ, and $p = \gamma q + 1$ for a specified integer γ. Then, an element f_3 of order q in the multiplicative group $\mathbb{Z}_p^*$ is defined. Secret key $x_{\mathcal{B}} \in \mathbb{Z}_q^*$ is created. The corresponding public key is (p, q, f_3, F), where $F = f_3^{x_{\mathcal{B}}} \pmod p$.

The Signature Scheme. Let $m \in G_q$ the message to be signed and $m' \in \{0,1\}^*$ another (possibly empty) message associated to the signature. Let $M = [m']\|m$.[9] The signature *Sig(M)* on M consists of $z = m^{x_{\mathcal{B}}} \pmod p$ along with a proof that $log_{f_3} F = log_m z$. So, we have

$$Sig(M) = (z, SK[\alpha : z = m^\alpha \wedge F = f_3^\alpha](m')) = (z, c, r)$$

The verifier of such a signature checks whether c is equal to:

$$c = \mathcal{H}(m'\|m\|z\|f_3\|F_3\|m^r z^c\|f_3^r F^c)$$

The Blind Signature Protocol. This signature scheme can be transformed into a blind signature scheme using Ohta-Okamoto's techniques [25]. To get a blind signature on the message m (of order q in $\mathbb{Z}_p^*$), one (the verifier) chooses a random $s \in \mathbb{Z}_q^*$ and asks the signer to sign $m_0 = m f_3^s$ (the input of the protocol). Let $z_0 = m_0^{x_{\mathcal{B}}} \pmod p$. The signer then proves (interactively with the verifier) that $\log_{f_3} F = \log_{m_0} z_0$. From this proof, the verifier can deduce the signature on m (see Figure 1). Following [13], we will denote this protocol *BlindSig(M, m_0)*. (For a discussion of the security of this protocol, we refer interested readers to [10,11]. In particular, it is shown in [11] that *BlindSig* satisfies the *correctness* and *blindness* requirements[10]).

The Coin Tracing Phase. Roughly, the user generates a 'verifiable' ElGamal encryption, computed with $\mathcal{T}$'s public key, of the value that will be used to blind the input of *BlindSig*.

[9] The notation $[m]$ means that m is optional.

[10] See [21] for a formal definition of blind signature schemes.

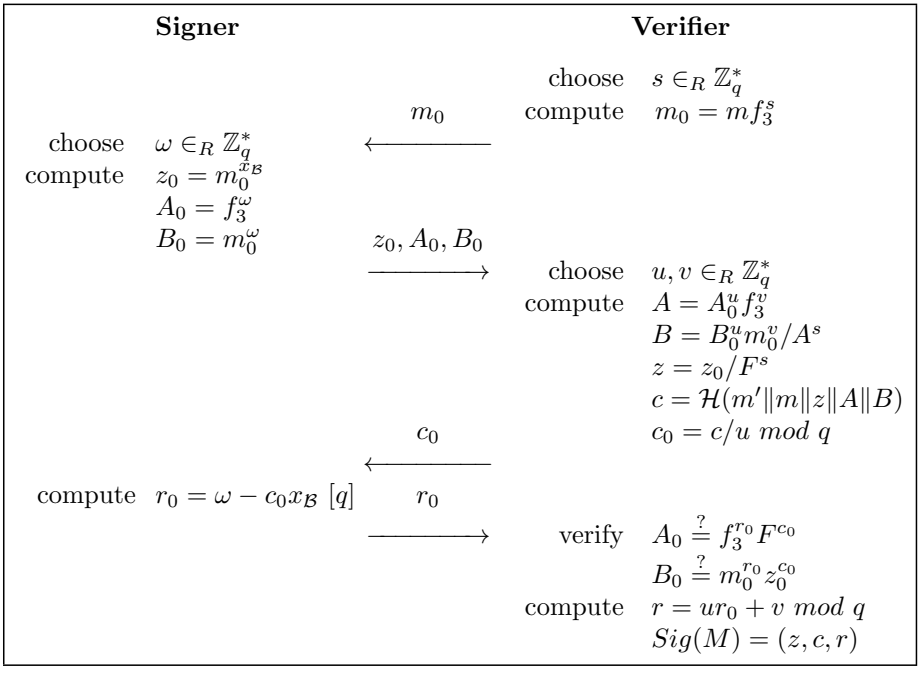

Fig. 1. The blind signature protocol *BlindSig(M,m_0)*

1. $\mathcal{U}$ authenticates itself to $\mathcal{B}$, so that $\mathcal{B}$ will be sure that $\mathcal{U}$ is the owner of the account I.
2. $\mathcal{U}$ randomly chooses $(s,t) \in \mathbb{Z}_q^*$, computes $E_1 = g_2^s f_3^t, E_2 = g_3^t$. E_1 and E_2 will be bound, as we will see, to the coin itself.
   Then $\mathcal{U}$ proves ($Proof_1$) to $\mathcal{B}$ (using the Chaum-Pedersen identification protocol [10]) that $\log_{f_3} E_1 = \log_{g_3} E_2$.

$$Proof_1 = SK\left[(\alpha,\beta) : E_2 = g_3^\alpha \wedge E_1 = g_2^\beta f_3^\alpha\right](\epsilon)$$

3. $\mathcal{B}$ verifies the proof and if the verification holds, stores the pair (E_1, E_2) in the user's entry of the withdrawal database for possible later anonymity revocation.[11]

The Withdrawal Phase

1. $\mathcal{U}$ randomly chooses $(a,b) \in_R \mathbb{Z}_q^*$ and computes the following message:

$$m' = A_2 \| D \| E \text{ where } A_2 = f_2^s,\ D = g_1^a g_2^b \text{ and } E = f_2^b.$$

[11] The pair (E_1, E_2) is an ElGamal encryption, computed with $\mathcal{T}$'s public key.

2. Both the user and the bank prepare the execution of the *BlindSig* protocol by computing independently the blinded coin $m_0 = I \times g_4 \times E_1$, where $\mathcal{B}$ has got E_1 from the coin tracing phase. Note that $m_0 = coin \times f_3^t$, where $coin = I \times g_4 \times g_2^s$ is the message to be (blindly) signed. The value g_4 is used to *restrict* the blind manipulations that the user can do. $\mathcal{U}$ and $\mathcal{B}$ execute the *BlindSig* protocol with m_0 as input. At the end of this protocol, $\mathcal{U}$ obtains a blind signature *Sig* on the message $(m' \| coin)$.
3. $\mathcal{B}$ debits the real money counterpart of the withdrawn coin from $\mathcal{U}$'s account.

The purpose of the message m' is to ensure the possibility of owner tracing and to make *double-spender* identification possible without the help of $\mathcal{T}$. We will see how this works below.

2.3 The Payment Protocol

We assume that the shop $\mathcal{S}$ is known under $Id_{\mathcal{S}}$ (its account number for example), and define "*DH*" to be the payment (date and) time. During the payment protocol, the user $\mathcal{U}$ sends *coin*, *Sig* and m' to $\mathcal{S}$. He then proves to $\mathcal{S}$ ($Proof_2$) that he knows the representation of $A_1 = coin/g_4$ with respect to (g_1, g_2) and that $log_{g_2} A_1 = log_{f_2} A_2$ (see Figure 2).

$$Proof_2 = SK\left[(\alpha, \beta) : A_2 = f_2^{\alpha} \wedge A_1 = g_1^{\beta} g_2^{\alpha}\right](Id_{\mathcal{S}} \| DH \| coin \| m')$$

$\mathcal{S}$ verifies the signature *Sig* and the proof and, if the verification holds, accepts the payment.

User		**Shop**
compute $r_1 \equiv a - cu \ [q]$		
$r_2 \equiv b - cs \ [q]$		
and $c = \mathcal{H}(Id_S \| DH \| coin \| m')$		
	$coin, Sig, m'$	
	c, r_1, r_2	
	$\longrightarrow$	verify
		$c \stackrel{?}{=} \mathcal{H}(Id_S \| DH \| coin \| m')$
		$D \stackrel{?}{=} g_1^{r_1} g_2^{r_2} A_1^c$
		$E \stackrel{?}{=} f_2^{r_2} A_2^c$
		$Proof_2 = (c, r_1, r_2)$

Fig. 2. Spending of the coin *coin*

2.4 The Deposit Protocol

To be credited the value of this coin, the shop sends the transcript of the execution of the payment protocol to the bank, which verifies, exactly as the shop did, that the coin (*coin*) bears the bank's signature and that the other responses are correct.

2.5 The Tracing Mechanisms

Double-spender Identification. If $\mathcal{U}$ spends the same coin more than once, for example in two different shops $\mathcal{S}$ and $\mathcal{S}'$, $\mathcal{B}$ will end up with values :

$$c = \mathcal{H}(Id_{\mathcal{S}} \| DH \| coin \| m'), \quad r_1 = a - cu \ mod \ q$$
$$c' = \mathcal{H}(Id_{\mathcal{S}'} \| DH' \| coin \| m'), r'_1 = a - c'u \ mod \ q$$

where Id_S and $Id_{S'}$ will differ. Thus, with high probability we have $c \neq c'(mod\ q)$, and $\mathcal{B}$ can compute

$$u = \frac{r'_1 - r_1}{c - c'} \ mod \ q$$

Owner Tracing. The trustee $\mathcal{T}$ is given the values A_1 and A_2 observed in a payment. $\mathcal{T}$ computes $A_1/{A_2}^{x_{\mathcal{T}}^{-1}} = g_1^u = I$ and thus obtains the account number of $\mathcal{U}$.

Coin Tracing. The trustee $\mathcal{T}$ is given a withdrawal transcript. $\mathcal{T}$ decrypts the ElGamal encryption (E_1, E_2) to obtain the value g_2^s. $\mathcal{T}$ then computes $coin = I \times g_4 \times g_2^s$. The coin *coin* can be put on a blacklist for recognizing it when it is spent.

3 On the Anonymity of [17]: Why Their Proof Is Incorrect?

We will explain why the proof of anonymity in [17] is incorrect.
For anonymity, they have to prove that the bank cannot solve the *GMWP* problem, i.e., the problem of linking withdrawal and payment transcripts (see also section 4 for a more formal definition of this problem). The data that are available for this linking are the following[12] (see [17] for more details).[13]

- At withdrawal : $[E_1^0 = g_2^{s_0} f_3^{m_0}, E_2^0 = g_3^{m_0}]$ and $[E_1^1 = g_2^{s_1} f_3^{m_1}, E_2^1 = g_3^{m_1}]$
- At payment : $[A_1^i = g_1^{u_i} g_2^{s_i}, A_2^i = f_2^{s_i}]$ and $[A_1^{\bar{i}} = g_1^{u_{\bar{i}}} g_2^{s_{\bar{i}}}, A_2^{\bar{i}} = f_2^{s_{\bar{i}}}]$
  $i, \bar{i} \in \{0, 1\}$, $i \neq \bar{i}$

where g_1, g_2, g_3, f_2, f_3 are public parameters (namely, four generators of a cyclic group G_q of prime order q) defined in the setup protocol, and s_0, s_1, m_0 and m_1 are random values defined by the users during the withdrawal protocol. u_i (resp. $u_{\bar{i}}$) is a secret value chosen by the user $\mathcal{U}_i$ (resp. $\mathcal{U}_{\bar{i}}$) during the account opening protocol. $g_1^{u_i}$ (resp $g_1^{u_{\bar{i}}}$) is $\mathcal{U}_i$'s (resp. $\mathcal{U}_{\bar{i}}$'s) account number. The linking problem is to determine whether i is 0 or 1.

Frankel et al. try to show that if there exists a machine $\mathcal{M}$ which given the information above can find i, then $\mathcal{M}$ can be used to break the ElGamal

[12] We voluntarily omit some values such as the proofs of knowledge V_1, V_2, V_3 and the bank's signature on the coins since they are not important for our analysis.
[13] We borrow their notation: X^i or X_i denotes value X at protocol i.

encryption in the sense of indistinguishability, i.e., break the DDH_q assumption [31]. Borrowing freely from the exposition in [17], we recall how their proof works. By definition of the security in the sense of indistinguishability, we get the freedom to choose two messages and then try to distinguish their encryptions. So let us choose two random values s_i and $s_{\bar{i}}$, and let $\mu_i = g_2^{s_i}$ and $\mu_{\bar{i}} = g_2^{s_{\bar{i}}}$ be these two messages. Let (E_1^0, E_2^0) and (E_1^1, E_2^1) be the encryptions of μ_0 and μ_1 respectively (where they implicitely assume that the ElGamal public key is in this case (g_3, f_3)). We are asked to distinguish these encryptions. We will use $\mathcal{M}$ for this purpose (step 2). We then feed $\mathcal{M}$ with these encryptions, plus (A_1^i, A_2^i) and $(A_1^{\bar{i}}, A_2^{\bar{i}})$, which we can construct for a random u_i, $u_{\bar{i}}$, since we know s_i and $s_{\bar{i}}$. The whole output of the simulator (consisting of the above values) is then fed to $\mathcal{M}$ which returns the value i, and thus break the ElGamal encryption in the sense of indistinguishability.

The problem with their proof comes from the fact that they omit to include in the bank's view of the withdrawal protocols the (public) account numbers $g_1^{u_i}$ and $g_1^{u_{\bar{i}}}$. (Indeed the bank has to know the user who withdraw a coin, since it will debit its account at the end of the withdrawal protocol). So the correct views of the withdrawal protocols are the following:[14]

- At withdrawal : $[E_1^0 = g_2^{s_0} f_3^{m_0}, E_2^0 = g_3^{m_0}, g_1^{u_0}]$
  $[E_1^1 = g_2^{s_1} f_3^{m_1}, E_2^1 = g_3^{m_1}, g_1^{u_1}]$
- At payment : $[A_1^i = g_1^{u_i} g_2^{s_i}, A_2^i = f_2^{s_i}]$
  $[A_1^{\bar{i}} = g_1^{u_{\bar{i}}} g_2^{s_{\bar{i}}}, A_2^{\bar{i}} = f_2^{s_{\bar{i}}}]$ for $i, \bar{i} \in \{0,1\}$, $i \neq \bar{i}$

(where for the reduction u_0 and u_1 are randomly chosen). So it seems much harder now given a pair of ElGamal ciphertexts to translate it into an instance of the *GMWP* problem. As we don't know i, how can we correctly construct the values A_1^i and $A_1^{\bar{i}}$? For example is A_1^i equal to $g_1^{u_0} g_2^{s_i}$ or to $g_1^{u_1} g_2^{s_i}$?[15] The reduction above will be correct if we manage to guess the correct value. But this is just what we are trying to determine! So we cannot feed $\mathcal{M}$ with correct payment transcripts and the reduction fails. Consequently, we cannot conclude that the *GMWP* problem implies the DDH_q problem.

4 Security

The security of the fair cash scheme, presented in section 2, can be described in three parts: (1) security for the shops and the bank (i.e., unreusability, unforgeability and unexpandability of coins; see [18] for a precise model), (2) security for $\mathcal{T}$ (i.e., the ability of $\mathcal{T}$ to trace), and (3) security for legitimate users (i.e., correctness and anonymity[16]). Since our main goal is to provide a provably anony-

[14] again, we voluntarily omit the values that are not necessary for our analysis.

[15] We don't have the freedom to choose u_0 and u_1 at step 2 (as this was done by Frankel et al.) since these values have been already defined in the withdrawal protocol.

[16] i.e., untraceability as defined in [18].

mous scheme, we engage in a detailed proof for the security for $\mathcal{U}$ (anonymity). The other proofs immediately follow from those of [13] (see also [17]).[17]
We know that in systems with an off-line trustee, $\mathcal{U}$'s anonymity can only be computational (see Theorem 1 of [16]). For our scheme, we will show that if anonymity is broken then the DDH_q assumption does not hold.

Theorem 1. *Under the DDH_q assumption and assuming the random oracle model, the above FOLC scheme satisfies the anonymity requirement.*

Proof (sketch). For anonymity, we have to prove that a collaboration of bank and shops cannot solve the following problem (see [18]):

Definition 1. *(General Matching Withdrawal-Payment (GMWP)): Given V_{W_0} and V_{W_1} the bank's views of two withdrawals W_0 and W_1 with the distinct users $\mathcal{U}_0$ and $\mathcal{U}_1$, C_r and $C_{\overline{r}}$, r, $\overline{r} \in \{0,1\}$, $r \neq \overline{r}$, the coins withdrawn in W_0 and W_1, P_r and $P_{\overline{r}}$ the corresponding transcripts of payments realized with C_r and $C_{\overline{r}}$, find r with probability non-negligibly better than random guessing (in n).*

We will first show that in fact the bank cannot link the coins obtained by the same user $\mathcal{U}$ to the corresponding executions of the withdrawal protocol. In other words, the bank cannot break the (seemingly weaker) following problem.

Definition 2. *(Matching Withdrawal-Payment (MWP)): Given V_{W_0} and V_{W_1} the bank's views of two withdrawals W_0 and W_1 with the user $\mathcal{U}$, C_r and $C_{\overline{r}}$, r, $\overline{r} \in \{0,1\}$, $r \neq \overline{r}$, the coins withdrawn in W_0 and W_1, P_r and $P_{\overline{r}}$ the corresponding transcripts of payments realized with C_r and $C_{\overline{r}}$, find r with probability non-negligibly better than random guessing (in n).*

Then we will show that in fact *GMWP* implies *MWP*.
Suppose we have a machine $\mathcal{M}$ that can break the above problem (*MWP*). Then we can use this machine as an oracle to break the ElGamal encryption in the sense of indistinguishability, i.e., break the DDH_q assumption, as follows (sketch). In order to show that the ElGamal encryption scheme is not secure in the sense of indistinguishability, it suffices to show that we can find, with non-negligible probability, a pair of plaintexts messages such that their encryptions can be distinguished with non-negligible probability of success.
Let p and q be two primes such that $|p-1| = \delta + n$ for a specified constant δ, and $p = \gamma q + 1$ for a specified integer γ. Let g_3 be a generator of G_q, the subgroup of order q of the multiplicative group $\mathbb{Z}_p^*$. Let $f_3 = g_3^{x_\mathcal{T}}$ be the public key of a party in the ElGamal encryption scheme and $x_\mathcal{T} \in_R \mathbb{Z}_q^*$ the corresponding private key. The bank secretly chooses $\delta \in_R \mathbb{Z}_q^*$, computes $g_2 = g_3^\delta \ mod\ p$ and publishes g_2. $\mathcal{B}$ then randomly chooses two integers s_0 and s_1 of $\mathbb{Z}_q$ and computes : $m_0 = g_2^{s_0} \ mod\ p$ and $m_1 = g_2^{s_1} \ mod\ p$. s_0, s_1, m_0, m_1 and g_2 are then published. Then given the ElGamal encryptions of m_0 and m_1 in a random order, i.e., $E^b = (E_b^1, E_b^2) = (g_2^{s_b} f_3^{t_b}, g_3^{t_b})$ and $E^{\overline{b}} = (E_{\overline{b}}^1, E_{\overline{b}}^2) = (g_2^{s_{\overline{b}}} f_3^{t_{\overline{b}}}, g_3^{t_{\overline{b}}})$ (for a

[17] The security for $\mathcal{S}$, $\mathcal{B}$ and $\mathcal{T}$ mainly relies on the assumption that the blind signature protocol that we use is a *restrictive* one (several well-known schemes, such as [4,11, 6,16,17], rely on the same assumption)

randomly chosen bit b), where t_b and $t_{\overline{b}} \in_R \mathbb{Z}_q^*$, we only need to show that given the machine $\mathcal{M}$, we can distinguish non-negligibly better than random guessing which ciphertext encrypts which message, i.e., find b.
To this effect we will first construct a (polynomial) converting algorithm AL which given the pair of ElGamal ciphertexts $(E^b,\ E^{\overline{b}})$ translates it into an instance of the MWP problem; hence if $\mathcal{M}$ can solve the MWP problem, it will break the ElGamal encryption in the sense of indistinguishability.

We remind that the bank's view V of a withdrawal and the transcript of a payment P consist of the following data:

- V: $I, E_1, E_2, Proof_1, c_0$ (the challenge sent by the user $\mathcal{U}$ during the *BlindSig* protocol).
- $P : coin, m' = A_2 \| D \| E, Sig = (z, c, r), Proof_2$.

Construction of AL : AL chooses $x_{\mathcal{B}} \in_R \mathbb{Z}_q^*$ and $g_4 \in_R G_q$ and computes $h_1 = g_1^{x_{\mathcal{B}}}$, $h_2 = g_2^{x_{\mathcal{B}}}$ and $h_4 = g_4^{x_{\mathcal{B}}}$ (where g_2, g_3, f_3 and δ has been defined before for the ElGamal encryption scheme). AL chooses $u \in_R \mathbb{Z}_q^*$ and computes $I = g_1^u$. AL also chooses random values for $Id_{\mathcal{S}}$ and DH (and also for $Id_{\overline{\mathcal{S}}}$ and $\overline{DH}$, introduced below) and two integers $(c_0^1, c_0^2) \in_R \mathbb{Z}_q^*$. AL then:

- simulates $Proof_{1b} = SK[(\alpha, \beta) : E_b^2 = g_3^\alpha \wedge E_b^1 = g_2^\beta f_3^\alpha]\,(\epsilon)$
  $Proof_{1\overline{b}} = SK[(\delta, \lambda) : E_{\overline{b}}^2 = g_3^\delta \wedge E_{\overline{b}}^1 = g_2^\lambda f_3^\delta]\,(\epsilon)$
- defines $V_b : I, E_b^1, E_b^2, Proof_{1b}, c_0^1$ and $V_{\overline{b}} : I, E_{\overline{b}}^1, E_{\overline{b}}^2, Proof_{1\overline{b}}, c_0^2$
  $A_1 = g_1^u g_2^{s_0}$, $A_2 = f_2^{s_0}$, $\hat{A}_1 = g_1^u g_2^{s_1}$, $\hat{A}_2 = f_2^{s_1}$
  (recall that AL knows u, s_0 and s_1)
- simulates $Proof_2 = SK[(\alpha, \beta) : A_2 = f_2^\alpha \wedge A_1 = g_1^\beta g_2^\alpha]\,(Id_{\mathcal{S}} \| DH)$
  $Proof_{\hat{2}} = SK[(\delta, \lambda) : \hat{A}_2 = f_2^\delta \wedge \hat{A}_1 = g_1^\lambda g_2^\delta]\,\left(Id_{\overline{\mathcal{S}}} \| \overline{DH}\right)$
- computes $z = (A_1 g_4)^{x_{\mathcal{B}}}$ and $\hat{z} = (\hat{A}_1 g_4)^{x_{\mathcal{B}}}$.
- simulates $Sig = SK[\alpha : z = (A_1 g_4)^\alpha \wedge F = f_3^\alpha]\,(A_2 \| D \| E)$
  $\overline{Sig} = SK[\delta : z = (\hat{A}_1 g_4)^\delta \wedge F = f_3^\delta](\hat{A}_2 \| \hat{D} \| \hat{E})$
  where D and E (resp. $\hat{D}$ and $\hat{E}$) are defined, as we will see during the simulation of $Proof_2$ (resp.$Proof_{\hat{2}}$).
- defines $P = \{A_1 g_4, Sig, A_2, D, E, Proof_2\}$
  $\hat{P} = \{\hat{A}_1 g_4, \overline{Sig}, \hat{A}_2, \hat{D}, \hat{E}, Proof_{\hat{2}}\}$

Simulations: It can be shown, using standard techniques, that in the random oracle model we can efficiently simulate the previous signatures of knowledge. Let us show for example how to simulate $Proof_2$ and Sig ($Proof_{1b}$, $Proof_{1\overline{b}}$ can be simulated in a very similar way).

Simulation of $Proof_2$ by AL:

- choose r_1, r_2, c at random in $\mathbb{Z}_q^*$
- compute $D = g_1^{r_1} g_2^{r_2} A_1^c$, $E = f_2^{r_2} A_2^c$
- define
  $\mathcal{H}(Id_{\mathcal{S}} \| DH \| A_1 g_4 \| A_2 \| D \| E) = c$
- return $Proof_2 = (c, r_1, r_2)$
  as the signature of knowledge

Simulation of Sig by AL:

- choose τ, $\widetilde{c}$ at random in $\mathbb{Z}_q^*$
- compute $A = f_3^{\tau}$, $B = (A_1 g_4)^{\tau}$
- define
  $\mathcal{H}(A_2 \| D \| E \| A_1 g_4 \| A \| B) = \widetilde{c}$
- compute $r = \tau - \widetilde{c} x_{\mathcal{B}} \bmod q$
- return $Sig = (z, \widetilde{c}, r)$
  as the signature of knowledge

V_b, $V_{\overline{b}}$, P and $\hat{P}$ are then fed to $\mathcal{M}$, which returns the value of b, and thus break the ElGamal encryption in the sense of indistinguishability. This concludes the first part of our proof (the bank cannot solve the MWP problem). □

Consider the following decision problem:

Definition 3. *Decision Withdrawal-Payment (DWP): Given V_W the view of $\mathcal{B}$ of a withdrawal W with the user $\mathcal{U}$, C a coin withdrawn and spent by $\mathcal{U}$, P the corresponding payment transcript, output 0 if C comes from W and 1 otherwise, with probability non-negligibly better than random guessing (in n).*

Fact 2 *The Decision Withdrawal-Payment problem implies the Matching Withdrawal-Payment problem.*

Clearly, the DWP problem is more difficult, since given a DWP oracle, one can simply test whether the coin C_r comes from W_0 or not. If the DWP oracle returns 0, then returns $r = 0$ and 1 otherwise.

By using the results of [19], we will show that $GMWP$ implies DWP. So we will have the following implications : $GMWP \Rightarrow DWP \Rightarrow MWP \Rightarrow DDH_q$. So under the DDH_q assumption, the bank cannot solve the $GMWP$ problem. This will conclude our proof.

Lemma 1. *The General Matching Withdrawal-Payment ($GMWP$) problem implies the Decision Withdrawal-Payment problem (DWP).*

Proof (sketch).
Note : For the sake of simplicity, we will ignore in the sequel, the signature of knowledge $Proof_1$ and the value c_0 in the bank's view of a withdrawal, since (as shown previously) we can easily simulate this proof and c_0 in the random oracle model. For the same reasons, we will ignore $Proof_2$, Sig and m' (which means that we ignore the payment transcripts). From now on, the view V_W of a withdrawal will consist of the data:

$V_W : g_1^u$ ($\mathcal{U}$'s account number), $g_2^s f_3^t, g_3^t$, for random values u, s, t.
A coin C is of the form : $g_1^u g_2^{\tilde{s}}, f_2^{\tilde{s}}$ (without loss of generality, we ignore g_4).

Definition 4. *We will represent a withdrawal view V_W by the triplet $[g_1^u, g_2^s f_3^t, g_3^t]$ and a coin C by the pair $[g_1^u g_2^{\tilde{s}}, f_2^{\tilde{s}}]$. We will call a pair (V_W, C) a WP-pair (where C is a coin withdrawn by $\mathcal{U}$ and V_W the bank's view of a withdrawal with $\mathcal{U}$). We will say that a WP-pair (V_W, C) is a correct pair if the coin C comes from the withdrawal W (which means that $s = \tilde{s}$ mod q, see above) and an incorrect pair otherwise (which means that $s \neq \tilde{s}$ mod q).*

Handschuh, Tsiounis and Yung have shown in [19] that matching problems imply decision problems provided that randomization of the input (the target instance of the decision problem) is possible[18] (see [19] for more details).[19]

Let $(V_W = [g_1^u, g_2^s f_3^t, g_3^t], C = [g_1^u g_2^{\tilde{s}}, f_2^{\tilde{s}}])$ be the target instance of the DWP problem. We can randomize this input as follows :

- choose : $w_1, w_2, w_3, w_4 \in_R \mathbb{Z}_q$
- compute : $V_W^R = [g_1^{uw_1} g_1^{w_4}, (g_2^s f_3^t)^{w_1} g_2^{w_2} f_3^{w_3}, (g_3^t)^{w_1} g_3^{w_3}]$
  $C^R = [(g_1^u g_2^{\tilde{s}})^{w_1} g_1^{w_4} g_2^{w_2}, f_2^{\tilde{s}w_1} f_2^{w_2}]$

It is now easy to see that if $s = \tilde{s}$ *mod* q then this randomization process generates all possible correct WP-pairs and if $s \neq \tilde{s}$ *mod* q, then the randomization process generates all the incorrect WP-pairs. Consequently, we are in the situation where the matching problem implies the decision problem and then the GMWP problem implies the DWP problem.

Using the techniques of section 4, it can be shown that [17] is secure in the MWP sense. Then, one can use our methodology for deriving the implication between $GMWP$ and MWP to prove that [17] is also secure in the $GMWP$ sense.

Acknowledgements. We would like to thank Marc Girault and Yiannis Tsiounis for fruitful discussions. We also thank Ari Juels for providing many useful comments.

[18] This is in fact necessary in the decision phase of their theorem 1.

[19] The full proof of lemma 1 follows in fact from the theorem 1 described in [19]. There are two phases in our proof: the testing phase and the decision phase. In the testing phase, the $GMWP$ oracle's behavior is tested. It can be shown that the $GMWP$ oracle can be used to distinguish either between two correct WP-pairs and one correct/one incorrect WP-pair, *or* between one correct/one incorrect WP-pair and two incorrect ones. In the decision phase, we can use the result of the testing phase to decide whether the target instance of the DWP problem is a correct WP-pair or not. The fact that the randomization of the input is possible is fundamental in this phase (in order to be able to feed the oracle with a randomize input sequence) (A detailed proof will appear in the full paper).

References

1. M. Abe and M. Ohkubo, Provably secure fair blind signatures with tight revocation, *Proceedings of ASIACRYPT'01*, Lecture Notes in Computer Science, vol. 2248, Springer-Verlag, pp. 583–601.
2. M. Bellare et P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp 62–73.
3. D. Boneh, The Decision Diffie-Hellman Problem, *Proceedings of the third Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science, vol. 1423, Springer-Verlag, pp. 48–63.
4. S. Brands, Untraceable Off-Line Cash in Wallets with Observers, *Proceedings of CRYPTO'93*, Lecture Notes in Computer Science, vol. 773, Springer-Verlag, pp. 302–318.
5. E. Brickell, P. Gemmel and D. Kravitz, Trustee-based tracing extensions to anonymous cash and the making of anonymous change, *Proceedings of the 6th Annual Symposium on Discrete Algorithm*, Jan 1995, pp. 457–466.
6. J. Camenisch, U. Maurer and M. Stadler, Digital payment systems with passive anonymity-revoking trustees, *Proceedings of ESORICS'96*, Lecture Notes in Computer Science, vol. 1146, Springer-Verlag, pp. 33–43.
7. J. Camenisch, U. Maurer and M. Stadler, Digital payment systems with passive anonymity-revoking trustees, Journal of Computer Security, vol. 5, number 1, IOS Press, 1997.
8. J. Camenisch, J.M. Piveteau and M. Stadler, An efficient fair payment system, *Proceedings of 3rd ACM Conference on Computer and Communications Security*, ACM Press, 1996, pp. 88–94.
9. J. Camenisch and M. Stadler, Efficient group signatures for large groups, *Proceedings of CRYPTO'97*, Lecture Notes in Computer Science, vol. 1296, Springer-Verlag, pp. 410–424.
10. D. Chaum and T. Pedersen, Wallet Databases with Observers, *Proceedings of CRYPTO'92*, Lecture Notes in Computer Science, vol. 740, Springer-Verlag, pp. 89–105.
11. R. Cramer and T. Pedersen, Improved privacy in wallets with observers, *Proceedings of EUROCRYPT'93*, Lecture Notes in Computer Science, vol. 765, Springer-Verlag, pp. 329–343.
12. G. Davida, Y. Frankel, Y. Tsiounis and M. Yung, Anonymity Control in E-Cash Systems, *Proceedings of Financial Cryptography'97*, Anguilla, British West Indies, vol. 1318, Springer-Verlag, pp. 1–16.
13. A. de Solages and J. Traoré, An Efficient Fair Off-Line Electronic Cash System with Extensions to Checks and Wallets with Observers, *Proceedings of Financial Cryptography'98*, vol. 1465, pp. 275–295.
14. T. El Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, IT-31, vol. 4, pp. 469–472, 1985.
15. A. Fiat and A. Shamir, How to Prove Yourself: Practical Solutions to Identification and Signature Problems, *Proceedings of CRYPTO'86*, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, pp. 186–194.
16. Y. Frankel, Y. Tsiounis and M. Yung, Indirect discourse proofs: achieving fair offline electronic cash, *Proceedings of ASIACRYPT'96*, Lecture Notes in Computer Science, vol. 1163, Springer-Verlag, pp. 244–251.

17. Y. Frankel, Y. Tsiounis and M. Young, Fair Off-Line e-cash Made Easy, *Proceedings of ASIACRYPT'98*, Lecture Notes in Computer Science, vol. 1514, Springer-Verlag, pp. 257-270.
18. M. Franklin and M. Yung, Secure and efficient off-line digital money, *Proceedings of ICALP'93*, Lecture Notes in Computer Science, vol. 700, Springer-Verlag, pp. 265–276.
19. H. Handschuh, Y. Tsiounis and M. Yung, Decision oracles are equivalent to Matching oracles, *Proceedings of PKC'99*, vol. 1560, pp. 276–289.
20. M. Jakobsson and M. Yung, Revokable and versatile electronic money, *Proceedings of 3rd ACM Conference on Computer and Communications Security*, ACM Press, 1996, pp. 76–87.
21. A. Juels, M. Luby and R. Ostrovsky, Security of blind digital signatures, *Proceedings of EUROCRYPT'97*, Lecture Notes in Computer Science, vol. 1294, Springer-Verlag, pp. 150–164.
22. A. Juels, Trustee tokens: simple and practical anonymous digital coin tracing, *Proceedings of Financial Cryptography'99*, Lecture Notes in Computer Science, vol. 1648, Springer-Verlag, pp. 29–45.
23. D. Kügler and H. Vogt, Off-line payments with auditable tracing, *Proceedings of Financial Cryptography'02*, Lecture Notes in Computer Science, Springer-Verlag.
24. L. Meier, Special aspects of escrowed-based e-cash systems, Master's Thesis, Universität des Saarlandes, March 2000.
25. T. Okamoto and K. Ohta, Divertible Zero-Knowledge Interactive Proofs and Commutative Random Self-Reducibility, *Proceedings of EUROCRYPT'89*, Lecture Notes in Computer Science, vol. 434, Springer-Verlag, pp. 481–496.
26. B. Pfitzmann and A.-R. Sadeghi, Self-escrowed cash against user blackmailing, *Proceedings of Financial Cryptography'00*, Lecture Notes in Computer Science, vol. 1962, Springer-Verlag, pp 42–52.
27. D. Pointcheval and J. Stern, Security proofs for signatures schemes, *Proceedings of EUROCRYPT'96*, Lecture Notes in Computer Science, vol. 1070, Springer-Verlag, pp. 387–398.
28. C.P. Schnorr, Efficient Signature Generation by Smart Cards, *Journal of Cryptology*, 4(3), pp. 161–174, 1991.
29. M. Stadler, J.M. Piveteau and J. Camenisch, Fair Blind Signatures, *Proceedings of EUROCRYPT'95*, Lecture Notes in Computer Science, vol. 921, Springer-Verlag, pp. 209–219.
30. J. Traoré, Group signatures and their relevance to privacy-protecting off-line electronic cash systems, *Proceedings of ACISP'99*, Lecture Notes in Computer Science, vol. 1587, Springer-Verlag, pp. 228–243.
31. Y. Tsiounis and M. Yung, On the security of El Gamal-based encryption, *Proceedings of PKC'98*, Lecture Notes in Computer Science, vol. 1431, Springer-Verlag, pp. 117–134.
32. S. von Solms and D. Naccache, On blind signatures and perfect crimes, Computer & Security, 11, 1992, pp. 581–583.

Appendix: Basic Key Techniques

In this section, we recall the background of the key techniques that are used in [13] as well as in this paper.

The Decision Diffie-Hellman Problem

In this section, we define the DDH problem in a group of prime order q (DDH_q). See [3] for a general and formal definition of this problem.

Definition 5. *(Decision Diffie-Hellman problem) For security parameter n, p a prime with $|p-1| = \delta + n$ for a specified constant δ, for $g \in \mathbb{Z}_p^*$ a generator of prime order $q = (p-1)/\gamma$ for a specified integer γ and for $(a,b) \in_R \mathbb{Z}_q$, given $\left[g^a (\mathrm{mod}\, p), g^b (\mathrm{mod}\, p), y\right]$ output 0 if $y \equiv g^{ab} (\mathrm{mod}\, p)$ and 1 otherwise, with probability non-negligibly better than random guessing (in n).*

The *Decision Diffie-Hellman assumption* states that it is infeasible for a probabilistic polynomial time adversary to solve the Decision Diffie-Hellman problem. [31] have proven that the semantic security of the ElGamal encryption [14] is equivalent to the Decision Diffie-Hellman assumption.

Signatures of Knowledge

These building blocks are signature schemes derived from 3-move honest-verifier zero-knowledge proofs of knowledge using the generic transformation introduced by Fiat and Shamir [15].[20] Following [9], we refer to such constructs as *signatures of knowledge*.

In the following we consider two such building blocks, borrowing some notation from [9]. They are constructed over a cyclic group G_q of prime order q. The first building block enables to show the knowledge and equality of two discrete logarithms of, say h_1 and h_2, with respect to the bases g_1 and g_2 (where g_1, g_2, h_1 and $h_2 \in G_q$), i.e., knowledge of an integer $x \in \mathbb{Z}_q$ satisfying $h_1 = g_1^x$ and $h_2 = g_2^x$ (cf. [10]).

Definition 6. *(Equality of discrete logarithms) Let g_1, g_2, h_1, $h_2 \in G_q$. A pair $(c,r) \in Z_q^2$ satisfying $c = \mathcal{H}(m \| g_1 \| h_1 \| g_2 \| h_2 \| g_1^r h_1^c \| g_2^r h_2^c)$ is a signature of knowledge of the discrete logarithm of both $h_1 = g_1^x$ with respect to g_1 and $h_2 = g_2^x$ with respect to g_2 on the message m.*

This signature is denoted by:[21]

$$SK[\alpha : h_1 = g_1^\alpha \wedge h_2 = g_2^\alpha](m)$$

Such a pair can be computed by a prover who knows the secret value x as follows: first choose a random value $a \in \mathbb{Z}_q$ and compute c and r as $c = \mathcal{H}(m \| g_1 \| h_1 \| g_2 \| h_2 \| g_1^a \| g_2^a)$ and $r = a - cx \mod q$. (In the sequel, we will deliberately omit the fixed values g_1, g_2, $h_1$and h_2 in the computation of c.)
The second building block is a signature of knowledge of the discrete logarithm

[20] Such signature schemes can be proven to be secure in the random oracle model [2] given the security of the underlying proof of knowledge [27].

[21] which can be read as a signature of knowledge of a value α such that $h_1 = g_1^\alpha$ and $h_2 = g_2^\alpha$ hold. The convention is that Greek letters denote the secret knowledge and the other letters denote public parameters between the signer and the verifier.

of h to the base g and of a representation of h_1 to the base (g_1, g_2), where the g_2-part of this representation equals the discrete logarithm of h to the base g. The signature on the message m is denoted by:

$$SK[(\alpha, \beta) : h = g^{\alpha} \wedge h_2 = g_1^{\beta} g_2^{\alpha}](m)$$

This building block easily derives from the previous one.

Retrofitting Fairness on the Original RSA-Based E-cash

Shouhuai Xu[1] and Moti Yung[2]

[1] Dept. of Information and Computer Science, University of California at Irvine
shxu@ics.uci.edu
[2] Dept. of Computer Science, Columbia University
moti@cs.columbia.edu

Abstract. The notion of *fair* e-cash schemes was suggested and implemented in the last decade. It balances anonymity with the capability of tracing users and transactions in cases of crime or misbehavior. The issue was raised both, in the banking community and in the cryptographic literature. A number of systems were designed with an off-line fairness, where the tracing authorities get involved only when tracing is needed. However, none of them is based on the original RSA e-cash. Thus, an obvious question is whether it is possible to construct an *efficient* fair e-cash scheme by retrofitting the fairness mechanism on the original RSA-based scheme. The question is interesting from, both, a practical perspective (since investment has been put in developing software and hardware that implement the original scheme), and as a pure research issue (since retrofitting existing protocols with new mechanisms is, at times, harder than designing solutions from scratch). In this paper, we answer this question in the affirmative by presenting an efficient fair off-line e-cash scheme based on the original RSA-based one.

Keywords. E-cash, Fairness, Conditional Anonymity, RSA.

1 Introduction

In his seminal paper [C82], David Chaum introduced the notion of e-cash and presented the first e-cash scheme based on RSA ([RSA78]) blind signature. This scheme prevents the users from double-spending e-coins by on-line checking their freshness against a list of e-coins that have been spent. The overhead of this on-line checking process had inspired the notion and first implementation of off-line e-cash scheme [CFN88], which is also based on RSA blind signature but ensures that the identity of a double-spender will be exposed (after the fact). The above schemes preserve information-theoretic anonymity for users who do not double-spend any e-coin; this leads to potential abuses as was initially discussed in [vSN92]. In the last decade, this vulnerability has inspired numerous *fair* (or, conditionally anonymous) off-line e-cash schemes, but none of them (see, for example, [CMS96,FTY96,JY96]) is based on the original RSA e-cash; instead,

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 51–68, 2003.
© Springer-Verlag Berlin Heidelberg 2003

they all use discrete logarithm based cryptosystems. Thus, it has been open whether it is possible to construct an *efficient* fair off-line e-cash scheme that retains the original RSA-based scheme. In this paper, we answer this question in the affirmative.

1.1 Our Contributions

We show how to incorporate *fairness* into the original RSA-based e-cash by presenting an *efficient* fair off-line RSA e-cash scheme. In the resulting scheme, fairness is implemented by deploying a set of servers that maintain a threshold cryptosystem such that (1) the servers stay off-line when nothing goes wrong (i.e., they are invoked only when there is a need to revoke anonymity), and (2) the servers are only assumed to preserve user anonymity. Our scheme preserves the system architecture of its underlying scheme, namely the Chaum-Fiat-Naor one [CFN88], which has been developed by companies and has been deployed experimentally in the real world. These facts make retrofitting fairness while retaining the system architecture important from a business perspective. Our scheme is flexible in the sense that, as a special case, it can retain the original Chaum-Fiat-Naor coin structure as is (for software compatibility reasons), while it also allows other desirable coin structures.

Beyond achieving the above solution, there is a crucial technical question related to our solution that we believe deserves special mention: In order to implement fairness, what properties should the cryptosystem possess (e.g., is it enough to deploy a standard semantically-secure cryptosystem like ElGamal)? Although it may seem intuitively sufficient to deploy the ElGamal cryptosystem, we actually show that it is not so in the context of fair e-cash where the adversary has additional capability in the possible fault scenario (see Section 3.3 for details).

Remark 1. The e-cash scheme due to Juels [J99] inherits the e-coin structure of [C82]. However, this scheme involves an *on-line* Trusted Third Party (TTP) in the withdrawal sessions; the use of such an on-line TTP follows [BGK95]. Moreover, this scheme assumes a trust model that is strictly stronger than its counterpart in our scheme: there the TTP, in addition to being liable for preserving anonymity, is also assumed not to frame a user or steal a user's money; in our scheme, the TTP (namely the revocation servers) is only liable for preserving anonymity (i.e., the revocation servers have no capability to frame users or steal users' money).

Remark 2. The distribution of the revocation capability among a set of servers is done in order to implement better anonymity, as was shown in [MP98].

Organization:

In Section 2 we present the model and some basic goals of e-cash schemes. In Section 3 we introduce the basic ideas underlying our approach and construction. In Section 4 we present the cryptographic tools that will be used to implement fairness. We present our fair off-line RSA e-cash scheme in Section 5, and analyze its properties in Section 6. We discuss some deployment issues and extensions in

Section 7 and conclude in Section 8. Due to space limitation, we have eliminated in the current version certain details which will be given in the full version of this paper.

2 The Model and Goals

The Participants. We consider the following entities: a bank that issues e-coins (which possess a structure we call the "coin structure"), a set of n revocation servers $\mathcal{P}_1, \cdots, \mathcal{P}_n$ that maintain a threshold decryption capability for revoking anonymity, a set of users that withdraw and spend e-coins, and a set of merchants that accept e-coins. All of the entities are modeled as probabilistic polynomial-time interactive Turing machines.

The Communication Channels. The communication channels, except those channels between the servers in the revocation process, are asynchronous. Regarding the communication channels between the servers in the revocation process, we assume that $\mathcal{P}_1, \cdots, \mathcal{P}_n$ are connected by a complete network of private (i.e., untappable) point-to-point channels, and that they have access to a dedicated broadcast channel. Furthermore, we assume a *fully synchronous* communication model such that messages of a given round in the protocol are sent by all parties simultaneously and are delivered to their recipients. However, all the results in this paper apply to the more realistic *partially synchronous* communication model, in which messages sent on either a point-to-point or the broadcast channel are received by their recipients within some fixed time bound. Note that when we deploy the system in the real world, these assumptions can be substituted with appropriate cryptographic protocols that enforce "rounds of communication" using commitment schemes before decommitment of actual values.

The Adversary. We consider a probabilistic polynomial-time adversary that may initiate various protocols. The adversary is t-threshold meaning that it is able to corrupt at most t revocation servers. The adversary is malicious in that it may cause the corrupt parties to arbitrarily divert from the specified protocol. We assume that the adversary is static, which means that it chooses the parties it will corrupt at the initialization of the system. (We defer to Section 7 a detailed discussion on the issue of a subtle "suicide" attack against the anonymity of the honest users; in this attack an element in the coin structure of a honest user is embedded into a coin structure of a dishonest user who will commit a crime in order to compromise the anonymity of the honest user. Such an attack requires knowledge of coin structure of other users and potentially involves a bank which does not keep secret the coin database.)

2.1 The Goals

We focus on the following basic goals of e-cash (presented informally):

1. Unforgeability: After initiating polynomially many withdrawal sessions, the adversary is still unable to output an e-coin that is different from all the e-coins obtained in the withdrawal sessions. In other words, no adversary can succeed in conducting a "one more coin forgery."
2. Revocability: The revocation servers can collaboratively expose the e-coin issued in a withdrawal session and associate a given e-coin with the corresponding withdrawal session.
3. Anonymity: An adversary succeeds in breaking anonymity if it can associate an e-coin with the corresponding withdrawal session initiated by a honest user, or associate two e-coins with the same honest user (although the user's identity may be unknown). We require the probability of an adversary successfully breaking anonymity to be negligible.

3 The Basic Ideas

In this section, we first recall Chaum-Fiat-Naor's off-line RSA e-cash scheme [CFN88]; this is necessary for understanding our approach. Then, we present the basic ideas underlying our approach.

3.1 The Chaum-Fiat-Naor RSA E-cash Scheme

This scheme consists of four protocols: The Initialization, The Withdrawal, The Payment, and The Deposit.

The Initialization Protocol. Let f and g be two-argument collision-free functions, where f behaves like a random oracle and g has the property that fixing the first argument gives a one-to-one or c-to-1 map from the second argument *onto* the range.

1. The bank initially publishes an RSA signature verification key $(3, N)$, where the modulus N is the multiplication of two prime numbers P and Q that are chosen according to the main security parameter κ.
2. The bank sets a secondary parameter l.
3. A user Alice opens a bank account numbered u and the bank keeps a counter v associated with it.

The Withdrawal Protocol. Let $||$ denote string concatenation. This protocol has the following steps.

1. Alice chooses a_i, c_i, d_i, and r_i, $1 \leq i \leq l$, independently and uniformly at random from Z_N^*.
2. Alice sends to the bank l blinded candidates $B_i = r_i^3 \cdot f(x_i, y_i) \bmod N$ for $1 \leq i \leq l$, where $x_i = g(a_i, c_i)$ and $y_i = g(a_i \oplus (u||(v+i)), d_i)$.
3. The bank chooses a random subset of $l/2$ blinded candidate indices $R = \{i_j\}_{1 \leq i_j \leq l, 1 \leq j \leq l/2}$ and transmits it to Alice.
4. Alice presents the a_i, c_i, d_i, and r_i for all $i \in R$, and the bank checks their semantical correctness (i.e., that their structure follows the protocol specification). To simplify notations, let $R = \{l/2+1, l/2+2, \cdots, l\}$.

5. The bank gives Alice $\prod_{1 \le i \le l/2} B_i^{1/3} \bmod N$ and charges her account one dollar. The bank also increments Alice's counter v by l.
6. Alice can easily extract the e-coin $C = \prod_{1 \le i \le l/2} f(x_i, y_i)^{1/3} \bmod N$. Alice re-indexes the candidates in C according to the f values: $f(x_1, y_1) < f(x_2, y_2) < \cdots < f(x_{l/2}, y_{l/2})$. Alice also increases her copy of the counter v by l.

The Payment Protocol. The payment protocol has the following steps.

1. Alice (anonymously) sends $C = \prod_{1 \le i \le l/2} f(x_i, y_i)^{1/3} \bmod N$ to Bob.
2. Bob chooses a random binary string $z_1, z_2, \cdots, z_{l/2}$.
3. For all $1 \le i \le l/2$: if $z_i = 1$, Alice sends Bob a_i, c_i, and y_i; otherwise, Alice sends Bob x_i, $a_i \oplus (u||(v+i))$, and d_i.
4. Bob verifies the semantical correctness of the responses.

The Deposit Protocol. The deposit protocol has the following steps.

1. Bob sends an e-coin C and Alice's responses in a payment to the bank.
2. The bank verifies their semantical correctness and credits Bob' account. The bank stores C, the binary string $z_1, z_2, \cdots, z_{l/2}$ and the values a_i, c_i, and y_i (for $z_i = 1$) and x_i, $a_i \oplus (u||(v+i))$, and d_i (for $z_i = 0$).

3.2 Key Observations Underlying Our Approach

An intuitive solution to incorporating *fairness* into the Chaum-Fiat-Naor scheme is to deploy a set of servers for revoking anonymity such that, ideally, the servers stay completely off-line (i.e., they are involved only when there is a need to revoke anonymity). The key observations are:

1. We can force a user to encrypt a coin representation using a public key whose corresponding private key is shared among the servers.
2. The revocation capability can be completely independent of the e-cash issuing capability, thus we can modularly integrate different cryptosystems (for different purposes) into a single scheme.

For concreteness, in the sequel we consider the case of DLOG-based cryptosystem for the servers (though other systems are possible). More specifically, we let a user encrypt $H_1(m) \in \mathcal{G}$, corresponding to which $H(m) \in Z_N^*$ appears in an e-coin, using a DLOG-based cryptosystem over a cyclic sub-group $\mathcal{G}$ of Z_p, where both $H(\cdot)$ and $H_1(\cdot)$ are appropriate functions. An intriguing question is: Can we employ the ElGamal cryptosystem [E85] for this purpose? Next we present a detailed analysis that shows that ElGamal is insufficient.

3.3 Why ElGamal Cryptosystem Is Insufficient for Our Purpose?

ElGamal cryptosystem, which has been proven to be semantically secure [TY98], may not be enough to facilitate the task of revoking anonymity. For the sake of proving anonymity, we need to present a simulator that is able to emulate a real-world system while embedding a Decision Diffie-Hellman (DDH) challenge

tuple. The key observation underlying the infeasibility is that the adversary is allowed to invoke the revocation process, which means: the simulator must somehow be able to decrypt ciphertexts generated by the adversary; otherwise, the adversary is able to distinguish a simulated system from the real-world system. (The rationale that we have to allow the adversary to invoke the revocation process can be justified by, for instance, the following scenario: if the adversary commits some suspicious activities then the revocation process will be invoked in the real world; the adversary misbehaves anyway, or simply for the purpose of distinguishing a simulation from the real-world system.) A simple solution to this problem is to deploy some cryptosystem that is simulatable while allowing the simulator to hold the private key.

One may suggest that we can seek to utilize the random oracle and the cut-and-choose technique (both of them are deployed anyway) so that the simulator not knowing the private key can get through. However, this intuition is incorrect. First, even if the simulator knows all the plaintexts that are output by a random oracle (namely $H_1(\cdot)$ in our setting), the adversary can arbitrarily plant a single plaintext as an element in an e-coin structure. In order to have a better understanding about this issue, let us consider the following simplified scenario: (1) the adversary asks the oracle at two points 0 and 1, which means that the simulator knows $H_1(0)$ and $H_1(1)$; (2) the adversary presents an ElGamal encryption of $H_1(b)$ where $b \in_R \{0,1\}$; (3) in the revocation process the simulator not knowing the private key needs to output b. This is exactly the game used in defining security of the cryptosystem, which means that if the simulator does not know the corresponding private key but is able to output the correct plaintext, then we can easily make use of the simulator to break the DDH assumption. It is easy to see that the adversary can always play such a game in the e-cash scheme, because it can always pass the cut-and-choose test with probability 0.5. Since the adversary can dynamically "open" values and play with them in the game itself, we have to cope with this more dynamic nature of adversarial behavior. To this end, the bare ElGamal cryptosystem seems insufficient.

4 Cryptographic Tools for Revoking Anonymity

Now, we review some DLOG-based cryptographic tools that will be used as subroutines in the rest of this paper. The reader familiar with these tools may skip the technical description.

The Setting. Let κ' be a security parameter of DLOG-based cryptosystems, corresponding to which two prime numbers p and q are chosen so that $q|(p-1)$. Suppose $g \in Z_p$ is of order q, which means that g specifies a unique cyclic subgroup $\mathcal{G}$ in which the DDH assumption is assumed to hold; namely, it is infeasible to distinguish a random tuple (g, h, u, v) of four independent elements in $\mathcal{G}$ from a random tuple satisfying $log_g u = log_h v$.

A Semantically-Secure Encryption Scheme. We adopt the semantically-secure encryption scheme appeared in [JL00], which is similar to the standard ElGamal scheme except that it uses two generators (to allow the more

dynamic adversary). Let $\langle p, q, g, h, y = g^{x_1} h^{x_2} \bmod p \rangle$ be a public key and $\langle p, q, g, h, x_1, x_2 \rangle$ be the corresponding private key, where $(x_1, x_2) \in_R Z_q^2$, g and h are two random elements in $\mathcal{G}$ such that nobody knows $log_g h$. To encrypt a message $M \in \mathcal{G}$, one chooses $k \in_R Z_q$ and computes the ciphertext $(g^k \bmod p, h^k \bmod p, y^k M \bmod p)$. To decrypt a ciphertext (α, β, γ), one computes $M = \gamma / \alpha^{x_1} \beta^{x_2} \bmod p$.

Proof of the Same Logarithm. The following protocol is a zero-knowledge proof that $log_g u = log_h v$. Let $DLOG(g, u) = DLOG(h, v)$ denote such a proof. Suppose $x = log_g u = log_h v \bmod q$.

1. The prover chooses $k \in_R Z_q$, computes $A = g^k \bmod p$ and $B = h^k \bmod p$, and sends (A, B) to the verifier.
2. The verifier chooses a challenge $c \in_R Z_q$, and sends it to the prover.
3. The prover sends the verifier $d = c \cdot x + k \bmod q$.
4. The verifier accepts if $g^d = u^c A \bmod p$ and $h^d = v^c B \bmod p$.

The Fiat-Shamir transformation [FS86] can make this protocol non-interactive.

Proof of the Same Representation. The following protocol allows a prover to prove $REP(g, h; y) = REP(\alpha, \beta; \gamma)$, where $REP(a, b; c)$ denotes the representation of c with respect to the pair of bases (a, b) [CP92]. Suppose $y = g^{x_1} h^{x_2} \bmod p$ and $\gamma = \alpha^{x_1} \beta^{x_2} \bmod p$.

1. The prover chooses $k_1, k_2 \in_R Z_q$, and computes $A = g^{k_1} h^{k_2} \bmod p$ and $B = \alpha^{k_1} \beta^{k_2} \bmod p$. The prover sends (A, B) to the verifier.
2. The verifier chooses $c \in_R Z_q$, and sends it to the prover.
3. The prover sends the verifier $d_1 = c \cdot x_1 + k_1 \bmod q$ and $d_2 = c \cdot x_2 + k_2 \bmod q$.
4. The verifier accepts if $g^{d_1} h^{d_2} = y^c A \bmod p$ and $\alpha^{d_1} \beta^{d_2} = \gamma^c B \bmod p$.

The Fiat-Shamir transformation [FS86] can make this protocol non-interactive.

Feldman-VSS(t). This protocol allows a dealer to share $s \in_R Z_q$ among a set of players $\{\mathcal{P}_1, \cdots, \mathcal{P}_n\}$ via a t-degree polynomial [F87]. As a consequence, $\mathcal{P}_i$ holds a share s_i such that $s \overset{(t+1,n)}{\longleftrightarrow} (s_1, \cdots, s_n)$.

1. The dealer chooses a random t-degree polynomial $f(z) = a_0 + a_1 z + \cdots + a_t z^t$ over Z_q such that $s = a_0$. It broadcasts $A_l = g^{a_l} \bmod p$ for $0 \leq l \leq t$, computes and secretly sends $s_j = f(j)$ to player $\mathcal{P}_j$ for $1 \leq j \leq n$.
2. $\mathcal{P}_j$ $(1 \leq j \leq n)$ verifies if $g^{s_j} = \prod_{l=0}^{t} (A_l)^{j^l} \bmod p$. We call this equation "Feldman verification equation". If the verification fails, $\mathcal{P}_j$ broadcasts a *complaint* against the dealer.
3. The dealer receiving a complaint from player $\mathcal{P}_j$ broadcasts s_j that satisfies the Feldman verification equation.
4. The dealer is *disqualified* if either there are more than t complaints in Step 2, or its answer to a complaint in Step 3 does not satisfy the Feldman verification equation.

Pedersen-VSS(t). This protocol allows a dealer to share a pair of secrets $(s, s') \in_R Z_q^2$ among a set of players $\{\mathcal{P}_1, \cdots, \mathcal{P}_n\}$ via two t-degree polynomials [P91]. This protocol also uses a pair of bases (g, h) so that $log_g h$ is unknown. As a consequence, $\mathcal{P}_i$ holds a pair of shares (s_i, s'_i) such that $s \overset{(t+1,n)}{\longleftrightarrow} (s_1, \cdots, s_n)$ and $s' \overset{(t+1,n)}{\longleftrightarrow} (s'_1, \cdots, s'_n)$.

1. The dealer chooses two random polynomials $f(z) = a_0 + a_1 z + \cdots + a_t z^t$ and $f'(z) = b_0 + b_1 z + \cdots + b_t z^t$ over Z_q. Let $s \stackrel{def}{=} a_0$ and $s' \stackrel{def}{=} b_0$. It broadcasts $C_l = g^{a_l} h^{b_l} \bmod p$ for $0 \le l \le t$, computes and secretly sends $s_j = f(j)$, $s'_j = f'(j)$ to $\mathcal{P}_j$ for $1 \le j \le n$.
2. $\mathcal{P}_j$ $(1 \le j \le n)$ verifies if $g^{s_j} h^{s'_j} = \prod_{l=0}^{t} (C_l)^{j^l} \bmod p$. We call this equation "PEDERSEN verification equation". If the verification fails, $\mathcal{P}_j$ broadcasts a *complaint* against the dealer.
3. The dealer receiving a complaint from $\mathcal{P}_j$ broadcasts the values s_j, s'_j that satisfy the PEDERSEN verification equation.
4. The dealer is *disqualified* if either there are more than t complaints in Step 2, or its answer to a complaint in Step 3 does not satisfy the PEDERSEN verification equation.

JOINT-PEDERSEN-RVSS(t). This protocol allows a set of players $\{\mathcal{P}_1, \cdots, \mathcal{P}_n\}$ to jointly generate a pair of secrets $(a, b) \in_R Z_q^2$ via a pair of t-degree polynomials. As a consequence, $\mathcal{P}_i$ holds a pair of shares (a_i, b_i) of (a, b) such that $a \stackrel{(t+1,n)}{\longleftrightarrow} (a_1, \cdots, a_n)$ and $b \stackrel{(t+1,n)}{\longleftrightarrow} (b_1, \cdots, b_n)$.

1. $\mathcal{P}_i$ $(1 \le i \le n)$, as a dealer, performs an instance of PEDERSEN-VSS(t) to share a pair of secrets $(a_{i0}, b_{i0}) \in_R Z_q^2$ such that $a_{i0} \stackrel{(t+1,n)}{\longleftrightarrow} (s_{i1}, \cdots, s_{in})$ and $b_{i0} \stackrel{(t+1,n)}{\longleftrightarrow} (s'_{i1}, \cdots, s'_{in})$.
2. $\mathcal{P}_i$ $(1 \le i \le n)$ builds the set of non-disqualified players $QUAL$. This is a unique global name depending on the broadcast information available to all of the honest players.
3. $\mathcal{P}_i$ $(1 \le i \le n)$ holds (a_i, b_i) so that $a_i = \sum_{j \in QUAL} s_{ji} \bmod q$ is its share of $a = \sum_{i \in QUAL} a_{i0} \bmod q$, and $b_i = \sum_{j \in QUAL} s'_{ji} \bmod q$ is its share of $b = \sum_{i \in QUAL} b_{i0} \bmod q$.

RAND-GEN(t). This protocol allows a set of n servers to collaboratively generate $g^a \bmod p$ such that $a \in_R Z_q$ and $a \stackrel{(t+1,n)}{\longleftrightarrow} (a_1, \cdots, a_n)$ [GJKR99].

1. The severs $\{\mathcal{P}_1, \cdots, \mathcal{P}_n\}$ execute JOINT-PEDERSEN-RVSS(t).
2. $\mathcal{P}_i$ $(i \in QUAL)$ exposes $y_i = g^{a_i} \bmod p$ as follows.
   a) $\mathcal{P}_i$ broadcasts $A_{il} = g^{a_{il}} \bmod p$ for $0 \le l \le t$.
   b) $\mathcal{P}_j$ $(1 \le j \le n)$ verifies if $g^{s_{ij}} = \prod_{l=0}^{t} (A_{il})^{j^l} \bmod p$. If the verification fails for some index i, $\mathcal{P}_j$ broadcasts a complaint against $\mathcal{P}_i$ by broadcasting s_{ij} and s'_{ij} that satisfy the PEDERSEN verification equation but not the FELDMAN verification equation.
   c) If there is a valid complaint against $\mathcal{P}_i$ (i.e., the broadcast s_{ij} and s'_{ij} satisfy the PEDERSEN verification equation but not the FELDMAN verification equation), the servers reconstruct and publish a_{i0}, $f_i(z)$, A_{il} for $0 \le l \le t$. Each server in $QUAL$ sets $y_i = A_{i0} = g^{a_{i0}} \bmod p$ and $y = \prod_{i \in QUAL} y_i \bmod p$.
   d) As a result, $\mathcal{P}_j$ holds its share $a_j = \sum_{i \in QUAL} s_{ij} \bmod q$ of a. Note that $g^{a_j} = \prod_{i \in QUAL} g^{s_{ij}} \bmod p$ for $1 \le j \le n$ are publicly known.

EXP-INTERPOLATE. Given a set of values $(v_1, \cdots, v_n)$ where $n \geq 2t+1$, if at most t of them are *null* and the remaining are of the form $g^{a_i} \bmod p$ where the a_i's lie on some t-degree polynomial $F(\cdot)$ over Z_q, then we can compute $g^{F(0)} = \prod_{i \in \Gamma}(v_i)^{\lambda_{i,\Gamma}} = \prod_{i \in \Gamma}(g^{a_i})^{\lambda_{i,\Gamma}}$, where Γ is a $(t+1)$-subset of the correct v_i's and the $\lambda_{i,\Gamma}$'s are the corresponding Lagrange interpolation coefficients. Let $v = \text{EXP-INTERPOLATE}(v_1, \cdots, v_n)$.

DOUBLE-EXP-INTERPOLATE. Suppose that $(u, v) \in \mathcal{G}^2$, and that (a_i, b_i) for $1 \leq i \leq n$ are the shares output in an instance of JOINT-PEDERSEN-RVSS(t) with polynomials $F(\cdot)$, $F'(\cdot)$ and bases g, h. Given a set of values $(\varphi_1, \cdots, \varphi_n)$ where $n \geq 2t+1$, if at most t of them are *null* and the remaining are of the form $u^{a_i} v^{b_i} \bmod p$, then we can compute

$$\begin{aligned}\varphi &= \prod_{i \in \Gamma}(\varphi_i)^{\lambda_{i,\Gamma}} = \prod_{i \in \Gamma}(u^{a_i} v^{b_i})^{\lambda_{i,\Gamma}} = \prod_{i \in \Gamma}(u^{a_i})^{\lambda_{i,\Gamma}}(v^{b_i})^{\lambda_{i,\Gamma}} \\ &= (\prod_{i \in \Gamma}(u^{a_i})^{\lambda_{i,\Gamma}})(\prod_{i \in \Gamma}(v^{b_i})^{\lambda_{i,\Gamma}}) = u^{F(0)} v^{F'(0)} \bmod p,\end{aligned}$$

where Γ is a $(t+1)$-subset of the correct φ_i's, and the $\lambda_{i,\Gamma}$'s are the Lagrange interpolation coefficients. Let $\varphi = \text{DOUBLE-EXP-INTERPOLATE}(\varphi_1, \cdots, \varphi_n)$.

5 A Fair Offline RSA E-cash Scheme

We assume both $H : \{0,1\}^* \to Z_N^*$ and $H_1 : \{0,1\}^* \to \mathcal{G}$ behave like random oracles [BR93], where N is an RSA modulus and $\mathcal{G}$ is a group over which a DLOG-based public key cryptosystem is defined (see details below). The scheme consists of five protocols: THE INITIALIZATION, THE WITHDRAWAL, THE PAYMENT, THE DEPOSIT, and THE REVOCATION.

5.1 The Initialization Protocol

This protocol has the following steps.

1. Given a security parameter κ, the bank generates a pair of RSA public and private keys $(\langle e, N\rangle, \langle d, N\rangle)$ (e.g., $e = 3$) such that $ed = 1 \bmod \phi(N)$. Let l be the secondary security parameter.
2. Given a security parameter κ', the n revocation servers $\{\mathcal{P}_1, \cdots, \mathcal{P}_n\}$ generate a pair of public and private keys $(\langle p, q, g, h, y\rangle, \langle p, q, g, h, x_1, x_2\rangle)$ where $y = g^{x_1} h^{x_2} \bmod p$. This may include the following steps.
   a) GENERATING (p, q, g). This is a standard process.
   b) GENERATING h. The revocation servers run a distributed coin-flipping protocol to generate $r \in_R Z_p^*$. Then, it is sufficient to set $h = r^{(p-1)/q} \bmod p$. This is so because if q^2 does not divide $p-1$ then h is a random element in the group generated by g.

c) GENERATING y. The revocation servers run JOINT-PEDERSEN-RVSS(t) to share a random pair of $(a, b) \in_R Z_q^2$. Let $y = g^{x_1}h^{x_2} \stackrel{def}{=} g^a h^b = \prod_{i \in QUAL} g^{a_{i0}} h^{b_{i0}} \bmod p$, which is publicly known. Note that $x_1 = a = \sum_{i \in QUAL} a_{i0} \bmod q$ and $x_2 = b = \sum_{i \in QUAL} b_{i0} \bmod q$, that $\mathcal{P}_i$ $(1 \leq i \leq n)$ holds its shares $[x_1]_i = a_i = \sum_{j \in QUAL} s_{ji} \bmod q$ and $[x_2]_i = b_i = \sum_{j \in QUAL} s'_{ji} \bmod q$, and that $g^{[x_1]_i} h^{[x_2]_i} = \prod_{j \in QUAL} g^{s_{ji}} h^{s'_{ji}} \bmod p$ for $1 \leq i \leq n$ are publicly known.

5.2 The Withdrawal Protocol

The withdrawal protocol (between Alice and the bank) goes as follows.

1. Alice chooses $r_i \in_R Z_N^*$, m_i, and $k_i \in_R Z_q$, computes blinded candidates $B_i = r_i^e \cdot H(m_i) \bmod N$ and encryptions
$$(\alpha_i = g^{k_i} \bmod p, \beta_i = h^{k_i} \bmod p, \gamma_i = H_1(m_i) \cdot y^{k_i} \bmod p)$$
for $1 \leq i \leq l$. Then, Alice sends $\{(B_i, \alpha_i, \beta_i, \gamma_i)\}_{1 \leq i \leq l}$ to the bank. Here we assume that the m_i's are signature verification keys without pinning down any concrete signature scheme; as having said before, a special instantiation is an one-time signature scheme as in the Chaum-Fiat-Naor scheme.
2. The bank chooses a random subset of $l/2$ indices $R = \{i_j\}_{1 \leq i_j \leq l, 1 \leq j \leq l/2}$, and sends R to Alice.
3. Alice sends $\{(r_i, m_i, k_i)\}_{i \in R}$ to the bank.
4. The bank checks the semantical correctness of the responses. To simplify notations, let $R = \{l/2+1, \cdots, l\}$. Now, the bank gives Alice $\prod_{i=1}^{l/2} B_i^{1/e} \bmod N$ and charges her account (for instance) one dollar.
5. Alice obtains the e-coin
$$C = (m_1, \cdots, m_{l/2}; \prod_{i=1}^{l/2} H(m_i)^{1/e} \bmod N).$$
Alice re-indexes the m's in C to be lexicographic on their representation: $H(m_1) < H(m_2) < \cdots < H(m_{l/2})$.

5.3 The Payment Protocol

Suppose there is an anonymous channel between a user, Alice, and a merchant, Bob. The payment protocol has the following steps.

1. Alice sends Bob an e-coin $C = (m_1, \cdots, m_{l/2}; \prod_{i=1}^{l/2} H(m_i)^{1/e} \bmod N)$, and signatures that can be verified using the verification keys $(m_1, \cdots, m_{l/2})$.
2. Bob verifies the semantical correctness of the e-coin and the signatures.

5.4 The Deposit Protocol

The deposit protocol (between Bob and the bank) goes as follows.

1. Bob sends the bank an e-coin C and the corresponding signatures that can be verified using the verification keys $(m_1, \cdots, m_{l/2})$.
2. The bank verifies the semantical correctness of the payment. If the e-coin C has not been deposited before, then it credits Bob's account; otherwise, the bank invokes THE OWNER REVOCATION PROTOCOL below.

5.5 The Revocation Protocols

We consider two types of revocations: The Coin Revocation Protocol and The Owner Revocation Protocol.

The Coin Revocation Protocol. This protocol enables the servers to compute the e-coin issued in a given withdrawal session. Given a withdrawal session with ciphertexts

$$(\alpha_i = g^{k_i} \bmod p, \beta_i = h^{k_i} \bmod p, \gamma_i = H_1(m_i) \cdot y^{k_i} \bmod p)$$

for $1 \le i \le l/2$, the servers only need to collaboratively decrypt the ciphertexts and publish the corresponding plaintexts: $H_1(m_1), \cdots, H_1(m_{l/2})$. The distributed decryption operation goes as follows.

1. For $1 \le i \le l/2$, server $\mathcal{P}_a$ $(1 \le a \le n)$ publishes $\delta_{i,a} = \alpha_i^{[x_1]_a} \beta_i^{[x_2]_a} \bmod p$. To ensure robustness, $\mathcal{P}_a$ also proves
$$REP(g, h; g^{[x_1]_a} h^{[x_2]_a}) = REP(\alpha_i, \beta_i; \delta_{i,a})$$
using a non-interactive or interactive proof (in the latter case the verifier could be a honest one played by the servers that collaboratively choose a random challenge).
2. For $1 \le i \le l/2$, the servers compute $H_1(m_i) = \gamma_i / \delta_i \bmod p$, where
$$\delta_i = \text{Double-EXP-Interpolate}(\delta_{i,1}, \cdots, \delta_{i,n}).$$

Once the plaintexts $H_1(m_1), \cdots, H_1(m_{l/2})$ are known, an e-coin with

$$C = (m'_1, \cdots, m'_{l/2}; \textstyle\prod_{i=1}^{l/2} H(m'_i)^{1/e} \bmod N)$$

matches the withdrawal session if $H_1(m_i) = H_1(m'_j)$ for some $1 \le i, j \le l/2$.

The Owner Revocation Protocol. This protocol enables the servers to compute the owner of a given e-coin. For this purpose, we let the servers compare a given e-coin, $C = (m'_1, \cdots, m'_{l/2}; \prod_{i=1}^{l/2} H(m'_i)^{1/e} \bmod N)$, with *each* candidate withdrawal session of publicly known $\{(\alpha_i, \beta_i, \gamma_i)\}_{1 \le i \le l/2}$. We say "the e-coin C matches the withdrawal session" if there exist $1 \le i, j \le l/2$ such that $REP(g, h; y) = REP(\alpha_i, \beta_i; \gamma_i / H_1(m'_j))$.

An intuitive solution to the problem of deciding whether a given e-coin matches a given withdrawal session is to let the servers conduct a distributed proof $REP(g, h; y) = REP(\alpha_i, \beta_i; \gamma_i / H_1(m'_j))$. However, this would leak the plaintext information. To avoid this information leakage, it is natural to inject randomness into the revocation process. For example, the servers could compute $\theta = g^z \bmod p$, $\delta = \gamma_i / H_1(m'_j) \bmod p$, and $\Delta = [(\alpha_i)^z]^{x_1} [(\beta_i)^z]^{x_2} \bmod p$ for some $z \in_R Z_q$, and conduct a distributed protocol attempting to prove $DLOG(g, \theta) = DLOG(\delta, [(\alpha_i)^z]^{x_1} [(\beta_i)^z]^{x_2})$. While this is reasonable, we find an even better way to facilitate the task, namely letting the servers compute and compare $\delta^z \stackrel{?}{=} \Delta$. Note that $\delta^z = \Delta$ (therefore, the revocation process outputs Yes) if and only if $REP(g, h; y) = REP(\alpha_i, \beta_i; \gamma_i / H_1(m'_j))$ except for a negligible probability.

Specifically, the following protocol loops for $1 \le i \le l/2$ and $1 \le j \le l/2$, and may halt when it outputs the first Yes (there are various alternatives depending on the system management policy that is beyond the scope of this paper). Note that the protocol, for the sake of clarifying the presentation, is not round-optimal, but it is easy to see that rounds 3-8 can be merged into 2 rounds.

1. Define $\delta = \gamma_i / H_1(m'_j) \bmod p$.
2. The servers execute RANDOM-GEN(t) to generate $\theta = g^z \bmod p$ so that $z \overset{(t+1,n)}{\longleftrightarrow} (z_1, \cdots, z_n)$, where $\theta_a = g^{z_a} \bmod p$, $1 \le a \le n$, are publicly known.
3. Server $\mathcal{P}_a$ $(1 \le a \le n)$ broadcasts $\sigma_a = \delta^{z_a} \bmod p$. To ensure robustness, $\mathcal{P}_a$ proves in zero-knowledge $DLOG(g, \theta_a) = DLOG(\delta, \sigma_a)$.
4. The servers compute $\sigma = \text{EXP-INTERPOLATE}(\sigma_1, \cdots, \sigma_n)$.
5. Server $\mathcal{P}_a$ $(1 \le a \le n)$ broadcasts $\mu_a = \alpha_i^{z_a} \bmod p$. To ensure robustness, $\mathcal{P}_a$ proves in zero-knowledge $DLOG(g, \theta_a) = DLOG(\alpha_i, \mu_a)$.
6. The servers compute $\mu = \text{EXP-INTERPOLATE}(\mu_1, \cdots, \mu_n)$.
7. Server $\mathcal{P}_a$ $(1 \le a \le n)$ broadcasts $\nu_a = \beta_i^{z_a} \bmod p$. To ensure robustness, $\mathcal{P}_a$ proves in zero-knowledge $DLOG(g, \theta_a) = DLOG(\beta_i, \nu_a)$.
8. The servers compute $\nu = \text{EXP-INTERPOLATE}(\nu_1, \cdots, \nu_n)$.
9. Server $\mathcal{P}_a$ $(1 \le a \le n)$ broadcasts $\Delta_a = \mu^{[x_1]_a} \nu^{[x_2]_a} \bmod p$. To ensure robustness, $\mathcal{P}_a$ proves in zero-knowledge $REP(g, h; \psi_a) = REP(\mu, \nu; \Delta_a)$, where $\psi_a = g^{[x_1]_a} h^{[x_2]_a} \bmod p$ is publicly known.
10. The servers compute $\Delta = \text{DOUBLE-EXP-INTERPOLATE}(\Delta_1, \cdots, \Delta_n)$.
11. If $\Delta = \sigma$, then the protocol outputs YES meaning that $H_1(m'_j)$ is the plaintext corresponding to the ciphertext $(\alpha_i, \beta_i, \gamma_i)$; otherwise, the protocol outputs NO meaning that $H_1(m'_j)$ is not the plaintext corresponding to the ciphertext $(\alpha_i, \beta_i, \gamma_i)$.

6 Properties of the Fair Offline RSA E-cash Scheme

Let cfn denote the original Chaum-Fiat-Naor scheme [CFN88], and folc denote the above fair off-line scheme. As we said before, we focus on unforgeability, revocability, and anonymity.

6.1 Unforgeability

We reduce the unforgeability of cfn to the unforgeability of folc. Due to space limitation, we leave formal analysis to the full version of this paper.

6.2 Revocability

Revocability of folc is ensured by the cut-and-choose technique which assures that one element in the coin structure is decryptable by the authorities. Specifically, as in cfn, the probability that a dishonest user passes the cut-and-choose verification without presenting any appropriately ciphertext is negligible.

6.3 Anonymity

We consider two types of anonymity. First, no adversary can link a withdrawal session initiated by a honest user to the corresponding e-coin. To prove this (in Theorem 1), we need to ensure that THE COIN REVOCATION PROTOCOL does not leak any significant information about the private key except the plaintext

(this is proved in Lemma 1), and that THE OWNER REVOCATION PROTOCOL leaks no information about the private key and no information about the corresponding plaintext in the case that $(\alpha_i, \beta_i, \gamma_i)$ is no encryption of $H_1(m'_j)$ (this is proved in Lemma 2). Second, no adversary can link two e-coins withdrawn by a honest user (i.e., *unlinkability*) even if the user is not identified (this is proved in Theorem 2).

Before we prove the theorems, let us recall that the adversary is static and t-threshold (namely, it corrupts at most t revocation servers). Without loss of generality, we assume that $\mathcal{P}_1, \cdots, \mathcal{P}_{t'}$ $(t' \leq t)$ were corrupted at initialization, and thus their internal states are known to the adversary.

Lemma 1. *Given a ciphertext (α, β, γ) that appears in a withdrawal session.* THE COIN REVOCATION PROTOCOL *does not leak any information about (x_1, x_2) but the corresponding plaintext $H_1(m) = \gamma/\alpha^{x_1}\beta^{x_2} \bmod p$.*

Proof. (sketch) We construct a polynomial-time algorithm $\mathcal{S}$ to simulate THE COIN REVOCATION PROTOCOL. Suppose $\mathcal{S}$ is given $H_1(m)$ and the shares of (x_1, x_2) held by the corrupt servers: $([x_1]_1, [x_2]_1), \cdots, ([x_1]_{t'}, [x_2]_{t'})$. Recall that $g^{[x_1]_b} h^{[x_2]_b} \bmod p$ for $1 \leq b \leq n$ are publicly known. Let $\delta = \gamma/H_1(m) \bmod p$.

1. $\mathcal{S}$ simulates the process that $\mathcal{P}_b$ $(1 \leq b \leq n)$ publishes $\delta_b = \alpha^{[x_1]_b}\beta^{[x_2]_b} \bmod p$, which is intended to guarantee $\delta =$ DOUBLE-EXP-INTERPOLATE$(\delta_1, \cdots, \delta_n)$. For this purpose $\mathcal{S}$ executes as follows.
   a) $\mathcal{S}$ chooses $a_0 \in_R Z_q$, computes $\delta' = \delta/\alpha^{a_0} \bmod p$ which can be understood as $\beta^{a'_0} \bmod p$ for some unknown a'_0. Note that $\delta = \alpha^{a_0}\beta^{a'_0} \bmod p$.
   b) Note that $A_0 \stackrel{def}{=} \alpha^{a_0}$, $f(1) \stackrel{def}{=} [x_1]_1, \cdots, f(t') \stackrel{def}{=} [x_1]_{t'}$, $f(t'+1) \stackrel{def}{=} [x_1]^*_{t'+1} \in_R Z_q, \cdots, f(t) \stackrel{def}{=} [x_1]^*_t \in_R Z_q$ uniquely determine a t-degree polynomial $f(\eta) = a_0 + a_1\eta + \cdots + a_t\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v = \alpha^{a_v} = (A_0)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\alpha^{[x_1]_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\alpha^{[x_1]^*_b})^{\lambda_{vb}}$ for $1 \leq v \leq t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\alpha^{[x_1]^*_b} \stackrel{def}{=} \alpha^{f(b)} = \prod_{v=0}^{t} (A_v)^{b^v}$ for $t+1 \leq b \leq n$.
   c) Note that $A'_0 \stackrel{def}{=} \sigma' = \beta^{a'_0}$ where a'_0 is unknown to $\mathcal{S}$, $f'(1) \stackrel{def}{=} [x_2]_1, \cdots, f'(t') \stackrel{def}{=} [x_2]_{t'}$, $f'(t'+1) \stackrel{def}{=} [x_2]^*_{t'+1} \in_R Z_q, \cdots, f'(t) \stackrel{def}{=} [x_2]^*_t \in_R Z_q$ uniquely determine a t-degree polynomial $f'(\eta) = a'_0 + a'_1\eta + \cdots + a'_t\eta^t \bmod q$. So, $\mathcal{S}$ computes $A'_v = \beta^{a'_v} = (A'_0)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\beta^{[x_2]_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\beta^{[x_2]^*_b})^{\lambda_{vb}}$ for $1 \leq v \leq t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\beta^{[x_2]^*_b} \stackrel{def}{=} \beta^{f'(b)} = \prod_{v=0}^{t} (A'_v)^{b^v}$ for $t+1 \leq b \leq n$.
   d) For $1 \leq b \leq t'$, $\mathcal{S}$ executes at the adversary's will (e.g., the adversary may ask $\mathcal{S}$ to deviate from the protocol); for $t'+1 \leq b \leq n$, $\mathcal{S}$ publishes $\delta_b = \alpha^{[x_1]^*_b}\beta^{[x_2]^*_b} \bmod p$ and proves $REP(g, h; g^{[x_1]_b} h^{[x_2]_b}) = REP(\alpha, \beta; \delta_b)$ using the corresponding simulator.
2. It is guaranteed that $\delta =$ DOUBLE-EXP-INTERPOLATE$(\delta_1, \cdots, \delta_n)$ and thus $H_1(m) = \gamma/\delta \bmod p$.

Lemma 2. *Given $H_1(m'_j)$ and a ciphertext $(\alpha_i, \beta_i, \gamma_i)$,* The Owner Revocation Protocol *does not leak any information about the secret (x_1, x_2) except the bit whether $(\alpha_i, \beta_i, \gamma_i)$ is an encryption of $H_1(m'_j)$. Furthermore, in the case the process outputs* YES*, $H_1(m_i)$ is of course publicly known; in the case the process outputs* NO*, no information about $H_1(m_i) = \gamma_i/(\alpha_i)^{x_1}(\beta_i)^{x_2}$ is leaked.*

Proof. (sketch) We construct a polynomial-time algorithm $\mathcal{S}$ to simulate The Owner Revocation Protocol. Specifically, suppose an instance of The Owner Revocation Protocol outputs $\theta = g^z$, $\delta = \gamma_i/H_1(m'_j)$, $\sigma = \delta^z$, $\mu = (\alpha_i)^z$, $\nu = (\beta_i)^z$, and $\Delta = \mu^{x_1}\nu^{x_2}$, where either $\Delta = \sigma$ (i.e., the instance outputs Yes) or $\Delta \neq \sigma$ (i.e., the instance outputs No). Let $H_1(m_i) = \gamma_i/(\alpha_i)^{x_1}(\beta_i)^{x_2}$. Note that in the case $\Delta \neq \sigma$, the leakage of information about z may result in the leakage of information about the plaintext $H_1(m_i)$ because $\sigma/\Delta = [H_1(m_i)/H_1(m'_j)]^z$. Note also that $([x_1]_1, [x_2]_1)$, $\cdots$, $([x_1]_{t'}, [x_2]_{t'})$ are given to the simulator $\mathcal{S}$. $\mathcal{S}$ executes as follows.

1. $\mathcal{S}$ computes $\delta = \gamma_i/H_1(m'_j) \bmod p$, which is the same as the one given to $\mathcal{S}$.
2. $\mathcal{S}$ emulates the execution of Random-GEN(t) to generate $\theta = g^z \bmod p$, which is given as an input. This is done by calling the simulator in Appendix A. As a consequence, it is guaranteed that $z \overset{(t+1,n)}{\longleftrightarrow} (z_1, \cdots, z_n)$, where $\theta_b = g^{z_b} \bmod p$ for $1 \leq b \leq n$ are publicly known.
3. To simulate the generation of $\sigma = \delta^z$, $\mathcal{S}$ executes as follows.
   a) Note that $A_0^* \overset{def}{=} \sigma = \delta^{a_0^*}$ where $a_0^* \overset{def}{=} z$, $f^*(1) \overset{def}{=} z_1$, $\cdots$, $f^*(t') \overset{def}{=} z_{t'}$, $f^*(t'+1) \overset{def}{=} z^*_{t'+1} \in_R Z_q$, $\cdots$, $f^*(t) \overset{def}{=} z_t^* \in_R Z_q$ uniquely determine a t-degree polynomial $f^*(\eta) = a_0^* + a_1^*\eta + \cdots + a_t^*\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v^* = \delta^{a_v^*} = (A_0^*)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\delta^{z_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\delta^{z_b^*})^{\lambda_{vb}}$ for $1 \leq v \leq t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\delta^{z_b^*} \overset{def}{=} \delta^{f^*(b)} = \prod_{v=0}^{t} (A_v^*)^{b^v}$ for $t+1 \leq b \leq n$.
   b) For $1 \leq b \leq t'$, $\mathcal{S}$ emulates $\mathcal{P}_b$ at the adversary's will; for $t'+1 \leq b \leq n$, $\mathcal{S}$ emulates $\mathcal{P}_b$ by broadcasting $\sigma_b = \delta^{z_b^*} \bmod p$ and proving $DLOG(g, \theta_b) = DLOG(\delta, \sigma_b)$ using the corresponding simulator.
4. It is guaranteed that $\sigma = \delta^z = \text{EXP-Interpolate}(\sigma_1, \cdots, \sigma_n)$.
5. To simulate the generation of $\mu = \alpha_i^z$, $\mathcal{S}$ executes as follows.
   a) Note that $A_0^* \overset{def}{=} \mu = \alpha_i^{a_0^*}$ where $a_0^* \overset{def}{=} z$, $f^*(1) \overset{def}{=} z_1$, $\cdots$, $f^*(t') \overset{def}{=} z_{t'}$, $f^*(t'+1) \overset{def}{=} z^*_{t'+1} \in_R Z_q$, $\cdots$, $f^*(t) \overset{def}{=} z_t^* \in_R Z_q$ uniquely determine a t-degree polynomial $f^*(\eta) = a_0^* + a_1^*\eta + \cdots + a_t^*\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v^* = \alpha_i^{a_v^*} = (A_0^*)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\alpha_i^{z_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\alpha_i^{z_b^*})^{\lambda_{vb}}$ for $1 \leq v \leq t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\alpha_i^{z_b^*} \overset{def}{=} \alpha_i^{f^*(b)} = \prod_{v=0}^{t} (A_v^*)^{b^v}$ for $t+1 \leq b \leq n$.
   b) For $1 \leq b \leq t'$, $\mathcal{S}$ emulates $\mathcal{P}_b$ at the adversary's will; for $t'+1 \leq b \leq n$, $\mathcal{S}$ emulates $\mathcal{P}_b$ by broadcasting $\mu_b = \alpha_i^{z_b^*} \bmod p$ and proving $DLOG(g, \theta_b) = DLOG(\alpha_i, \mu_b)$ using the corresponding simulator.
6. It is guaranteed that $\mu = \alpha_i^z = \text{EXP-Interpolate}(\mu_1, \cdots, \mu_n)$.
7. To simulate the generation of $\nu = \beta_i^z$, $\mathcal{S}$ executes as follows.

a) Note that $A_0^* \stackrel{def}{=} \nu = \beta_i^{a_0^*}$ where $a_0^* \stackrel{def}{=} z$, $f^*(1) \stackrel{def}{=} z_1, \cdots, f^*(t') \stackrel{def}{=} z_{t'}$, $f^*(t'+1) \stackrel{def}{=} z^*_{t'+1} \in_R Z_q, \cdots, f^*(t) \stackrel{def}{=} z_t^* \in_R Z_q$ uniquely determine a t-degree polynomial $f^*(\eta) = a_0^* + a_1^*\eta + \cdots + a_t^*\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v^* = \beta_i^{a_v^*} = (A_0^*)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\beta_i^{z_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\beta_i^{z_b^*})^{\lambda_{vb}}$ for $1 \le v \le t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\beta_i^{z_b^*} \stackrel{def}{=} \beta_i^{f^*(b)} = \prod_{v=0}^{t} (A_v^*)^{b^v}$ for $t+1 \le b \le n$.

b) For $1 \le b \le t'$, $\mathcal{S}$ emulates $\mathcal{P}_b$ at the adversary's will; for $t'+1 \le b \le n$, $\mathcal{S}$ emulates $\mathcal{P}_b$ by broadcasting $\nu_b = \beta_i^{z_b^*} \bmod p$ and proving $DLOG(g, \theta_b) = DLOG(\beta_i, \nu_b)$ using the corresponding simulator.

8. It is guaranteed that $\nu = \beta_i^z = \text{EXP-INTERPOLATE}(\nu_1, \cdots, \nu_n)$.
9. In order to simulate the generation of $\Delta = \mu^{x_1}\nu^{x_2}$, $\mathcal{S}$ executes as follows.

a) $\mathcal{S}$ chooses $a_0 \in_R Z_q$, computes $\sigma' = \Delta/\mu^{a_0} \bmod p$ which can be understood as $\nu^{a_0'} \bmod p$ for some unknown a_0'. Note that $\Delta = \mu^{a_0}\nu^{a_0'} \bmod p$.

b) Note that $A_0 \stackrel{def}{=} \mu^{a_0}$, $f(1) \stackrel{def}{=} [x_1]_1, \cdots, f(t') \stackrel{def}{=} [x_1]_{t'}$, $f(t'+1) \stackrel{def}{=} [x_1]^*_{t'+1} \in_R Z_q, \cdots, f(t) \stackrel{def}{=} [x_1]_t^* \in_R Z_q$ uniquely determine a t-degree polynomial $f(\eta) = a_0 + a_1\eta + \cdots + a_t\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v = \mu^{a_v} = (A_0)^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\mu^{[x_1]_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\mu^{[x_1]_b^*})^{\lambda_{vb}}$ for $1 \le v \le t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\mu^{[x_1]_b^*} \stackrel{def}{=} \mu^{f(b)} = \prod_{v=0}^{t} (A_v)^{b^v}$ for $t+1 \le b \le n$.

c) Note that $A_0' \stackrel{def}{=} \sigma' = \nu^{a_0'}$ where a_0' is unknown to $\mathcal{S}$, $f'(1) \stackrel{def}{=} [x_2]_1, \cdots, f'(t') \stackrel{def}{=} [x_2]_{t'}$, $f'(t'+1) \stackrel{def}{=} [x_2]^*_{t'+1} \in_R Z_q, \cdots, f'(t) \stackrel{def}{=} [x_2]_t^* \in_R Z_q$ uniquely determine a t-degree polynomial $f'(\eta) = a_0' + a_1'\eta + \cdots + a_t'\eta^t \bmod q$. So, $\mathcal{S}$ computes $A_v' = \nu^{a_v'} = (A_0')^{\lambda_{v0}} \cdot \prod_{b=1}^{t'} (\nu^{[x_2]_b})^{\lambda_{vb}} \cdot \prod_{b=t'+1}^{t} (\nu^{[x_2]_b^*})^{\lambda_{vb}}$ for $1 \le v \le t$, where the λ_{vb}'s are the Lagrange interpolation coefficients. Finally, $\mathcal{S}$ computes $\nu^{[x_2]_b^*} \stackrel{def}{=} \nu^{f'(b)} = \prod_{v=0}^{t} (A_v')^{b^v}$ for $t+1 \le b \le n$.

d) For $1 \le b \le t'$, $\mathcal{S}$ executes according to the adversary's requirements (e.g., the adversary may ask $\mathcal{S}$ to deviate from the protocol); for $t'+1 \le b \le n$, $\mathcal{S}$ publishes $\Delta_b = \mu^{[x_1]_b^*}\nu^{[x_2]_b^*}$ and proves $REP(g, h; g^{[x_1]_b}h^{[x_2]_b}) = REP(\mu, \nu; \mu^{[x_1]_b^*}\nu^{[x_2]_b^*})$ using the corresponding simulator.

10. It is guaranteed that $\Delta = \text{DOUBLE-EXP-INTERPOLATE}(\Delta_1, \cdots, \Delta_n)$.
11. If $\Delta = \sigma$, then $\mathcal{S}$ outputs YES; otherwise, $\mathcal{S}$ outputs NO.

Theorem 1. *Suppose the DDH assumption holds. Then, no t-threshold adversary, who is given a ciphertext (α, β, γ) that appears in a withdrawal session initiated by a honest user, and $H_1(m)$, is able to correctly decide whether (α, β, γ) is an encryption of $H_1(m)$ with non-negligible advantage over a random guess.*

The proof will be given in the full version of this paper.

Theorem 2. *(unlinkability)* folc *is unlinkable meaning that no t-threshold adversary can link two e-coins to the same honest user (although the user's identity is unknown).*

Proof. (sketch) This is so because (1) all the m_i's chosen by the honest users are independently and uniformly distributed, and (2) Theorem 1 showed that folc, compared with cfn, does not leak any computational information about the secret (x_1, x_2) or the $H_1(m)$'s of the honest users'.

7 Discussions and Extensions

On the Efficiency of the Fair Off-Line RSA E-Cash Scheme. The penalty for fairness is the computational overhead at the user side, namely a user needs to compute $3l$ exponentiations for each e-coin (but all the exponentiations are pre-computable). The communication overhead at the user side is almost the same as in the Chaum-Fiat-Naor scheme. The time complexity of deciding whether an e-coin matches a withdrawal session is $O(l^2)$ comparison operations; this may be no real concern because that the servers deployed in threshold cryptosystems are typically powerful, and that the technique developed in [JM99] could be adopted to improve the revocation efficiency.

On Anonymity against "Suicide" Attacks. In the analysis of Section 6, we have taken into consideration the attack that the adversary attempts to distinguish a simulation from the real-world system by invoking the revocation process. Here we take a further step in investigating the possibility that the adversary attempts to break the anonymity of a honest user by conducting the following "suicide" attack. Suppose an adversary intercepts (from a withdrawal session initiated by a honest user) a ciphertext $(\alpha = g^k \bmod p, \beta = h^k \bmod p, \gamma = H_1(m) \cdot y^k \bmod p)$ that remains un-opened after the cut-and-choose verification. Furthermore, the adversary initiates a withdrawal session into which it embeds $(\alpha \cdot g^{k'} \bmod p, \beta \cdot h^{k'} \bmod p, \gamma \cdot y^{k'} \bmod p)$ as the i^{th} component ciphertext, where $i \in_R \{1, \cdots, l\}$. Note in this case the adversary passes the cut-and-choose verification with probability 0.5, which is enough. Then, the adversary commits some suspicious activities so that The Coin Revocation Protocol will be invoked and $H_1(m)$ will be publicly known. As a consequence, the adversary is able to associate the honest user's payment with her withdrawal. Note that this issue is typically ignored in e-cash protocols, although such a "suicide" attack against a honest user's anonymity is expensive because the adversary, whose account has been charged, can not spend that e-coin.

One may suggest that the above "suicide" attack can be blocked by making the communication channel in the withdrawal protocol confidential (e.g., using an appropriate cryptosystem). Clearly, a simple-minded encryption so that each candidate $(B_i, \alpha_i, \beta_i, \gamma_i)$ is individually encrypted using (for example) the bank's public key does not prevent an adversary from applying the cut-and-paste technique. Even more involved protocol (e.g., each withdrawal session could be protected by deploying a fresh session key generated using an authenticated key exchange protocol) does not completely solve the problem because the model, which reflects the well-known security principle called *separation-of-duty*, implies that the bank is not necessarily trusted in preserving the users' anonymity. Clearly, it does not help us to let the bank distribute the decryption capabili-

ties (corresponding to the fresh session keys) among a set of servers, since these servers are under the control of the bank.

In summary, the powerful but *expensive* "suicide" attack, which perhaps involves the bank that is not necessarily trusted in preserving the users' anonymity, is able (in some extreme cases as above) to compromise the anonymity of some honest users in the fair off-line RSA e-cash scheme, but anonymity of most honest users will be preserved.

8 Conclusion

We showed how to incorporate fairness into the Chaum-Fiat-Naor e-cash scheme while preserving their system architecture. The disadvantage of our scheme is the computational overhead at the user side; this may be solvable by deploying certain RSA-based (instead of DLOG-based) cryptosystem.

Acknowledgement. We thank the anonymous reviewers for helpful comments.

References

[BR93] M. Bellare and P. Rogaway. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. ACM CCS'93.

[BNPS01] M. Bellare, C. Namprempre, D. Pointcheval, M. Semanko. The Power of RSA Inversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme. Financial Crypto'01.

[BGK95] E. Brickell, P. Gemmell, and D. Kravitz. Trustee-based Tracing Extentions to Anonymous Cash and the Making of Anonymous Change. SODA'95.

[C82] D. Chaum. Blind Signatures for Untraceable Payments. Crypto'82.

[CFN88] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. Crypto'88.

[CMS96] J. Camenisch, U. Maurer, and M. Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trusrees. ESORICS'96.

[CP92] D. Chaum and T. Pedersen. Wallet Databases with Observers. Crypto'92.

[E85] T. El Gamal. A Public-Key Cryptosystem and a Signature Scheme Based on the Discrete Logarithm. IEEE Trans. IT, 31(4), 1985, pp 469–472.

[F87] P. Feldman. A Practical Scheme for Non-Interactive Verifiable Secret Sharing. FOCS'87.

[FS86] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. Crypto'86.

[FTY96] Y. Frankel, Y. Tsiounis, and M. Yung. Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash. Asiacrypt'96.

[FR95] M. Franklin and M. Reiter. Verifiable Signature Scharing. Eurocrypt'95.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Eurocrypt'99.

[GMR88] S. Goldwasser, S. Micali, R. Rivest. A Digital Signature Scheme Secure against Adaptive Chosen-message Attacks. SIAM J. Computing, 17(2), 1988.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game-A Completeness Theorem for Protocol with Honest Majority. STOC'87.

[JL00] S. Jarecki and A. Lysyanskaya. Concurrent and Erasure-Free Models in Adaptively-Secure Threshold Cryptography. Eurocrypt'00.
[JM99] M. Jakobsson and J. Mueller. Improved Magic Ink Signatures Using Hints. Financial Crypto'99.
[JY96] M. Jakobsson and M. Yung. Revokable and Versatile Electronic Money. ACM CCS'96.
[J99] A. Juels. Trustee Tokens: Simple and Practical Tracing of Anonymous Digital Cash. Financial Crypto'99.
[MP98] D. M'Raihl and D. Pointcheval. Distributed Trustees and Revocability: A Framework for Internet Payment. Financial Crypto'98.
[P91] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. Crypto'91.
[PS00] D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. J. of Cryptology, 13(3), 2000.
[R98] T. Rabin. A Simplified Approach to Threshold and Proactive RSA. Crypto'98.
[RSA78] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. CACM, 21(2), 1978, pp 120–126.
[S00] V. Shoup. Practical Threshold Signatures. Eurocrypt'00.
[TY98] Y. Tsiounis and M. Yung. On the Security of ElGamal Based Encryption. PKC'98.
[vSN92] S. von Solms and D. Naccache. On Blind Signatures and Perfect Crimes. Computer and Security, 11, 1992, pp 581–583.

A The Simulator of RANDOM-GEN(t)

This algorithm appeared in [GJKR99]. Given $\theta = g^z \bmod p$, the simulator emulates the generation of θ in the presence of an adversary that corrupts no more than t servers. Denote by $\mathbb{B}$ the set of servers corrupted by the adversary, and by $\mathbb{G}$ the set of honest server (run by the simulator). Without loss of generality, let $\mathbb{B} = \{1, \cdots, t'\}$ and $\mathbb{G} = \{t'+1, \cdots, n\}$, where $t' \leq t$.

INPUT: $\theta = g^z \bmod p$ and parameters (p, q, g, h).

The simulation goes as follows.

1. The simulator executes JOINT-PEDERSEN-RVSS(t) on behalf of the servers in $\mathbb{G}$.
2. The simulator emulates the extraction of $\theta = g^z$.
   - Compute $A_{il} = g^{z_{il}} \bmod p$ for $i \in QUAL \backslash \{n\}$ and $0 \leq l \leq t$.
   - Set $A^*_{n0} = \theta / \prod_{i \in QUAL \backslash \{n\}} A_{i0} \bmod p$.
   - Assign $s^*_{nj} = s_{nj}$ for $1 \leq j \leq t$.
   - Compute $A^*_{nl} = (A^*_{n0})^{\lambda_{l0}} \cdot \prod_{i=1}^{t} (g^{s^*_{ni}})^{\lambda_{li}} \bmod p$ for $1 \leq l \leq t$, where the λ_{li}'s are the Lagrange interpolation coefficients.
     a) Broadcast $\{A_{il}\}_{0 \leq l \leq t}$ for $i \in \mathbb{G} \backslash \{n\}$, and $\{A^*_{nl}\}_{0 \leq l \leq t}$.
     b) Check the values $\{A_{il}\}_{i \in \mathbb{B}, 0 \leq l \leq t}$ using the FELDMAN verification equation. If the verification fails for some $i \in \mathbb{B}$, $j \in \mathbb{G}$, broadcast a complaint (s_{ij}, s'_{ij}).
     c) If necessary, reconstruct the polynomial f_i and publish a_{i0} and $\theta_i = g^{a_{i0}} \bmod p$, where $i \in \mathbb{B}$.

Does Anyone Really Need MicroPayments?

Nicko van Someren (Moderator)[1],
Andrew Odlyzko[2], Ron Rivest[3], Tim Jones[4], and Duncan Goldie-Scot[5]

[1] CTO, nCipher Plc.
[2] University of Minnesota
[3] Founder, PepperCoin Inc.
[4] Founder, Mondex Plc.
[5] Editor, E-Finance Magazine

Abstract. Many cryptographers have tried to develop special technology for transferring tiny amounts of value; the theory being that the computational and/or administrative costs of other payment schemes render them unsuitable for small value transactions. In this panel we discussed two major questions: firstly are the existing systems really not useful for small values and secondly might other models such as flat rate or subscription systems be more suitable anyway, and be possible without the need for small payments?

1 Introduction

This panel session set out to examine the failure so far for any MicroPayment scheme to take off and asked if the reason for this might be that such schemes were not actually needed. Panellists were asked to provide their perspective on two main issues. Firstly, are existing payment schemes actually good enough for handing small value payments and thus there is no technical need for different, small value oriented systems. Secondly, were there in practice other payment options such as subscription models or aggregated payments that obviated the need to ever handle small value transactions.

The four panellists came from a variety of backgrounds, in the hope offering a wide set of perspectives. Andrew Odlyzko has published extensively on issues surrounding payments both while at AT&T and as a professor at the University of Minnesota. Ron Rivest, as well as co-inventing the RSA algorithm and being a professor at MIT has also designed a number of MicroPayments protocols and is one of the founders of PepperCoin, a company that plans to deliver a stochastic based MicroPayments system. Tim Jones was the founder and CEO of Mondex, a digital cash system based on stored value smart cards which allowed for transactions at essentially zero marginal cost and thus is suitable for certain types of MicroPayment problems. Duncan Goldie-Scot is a journalist who has been observing the digital money space for more than a decade and has written extensively on the rise and fall of various payment systems.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 69–76, 2003.
© Springer-Verlag Berlin Heidelberg 2003

2 Andrew Odlyzko

Andrew Odlyzko opened the discussion by stating that he had been working on MicroPayments for many years and that he hoped that some of the patents he held in this space might one day be worth something. That said, he presented a fairly pessimistic view on the prospects for the technology and supported this with four "Fundamental Reasons MicroPayments Will Never Happen".

The first reason he raised was that uses take a long time to accept new payment devices, especially if there is specific coercion to make a change. He illustrated this with the US dollar coin. Asserting that most observers outside the US are unaware of the fact that the US has a dollar coin at all he mentioned that failure of the "Susan B. Anthony" one dollar coin which had failed to gain widespread use. The more recent "gold" dollar coin has had such limited use that only three of the dozens of American audience members had made use of one. Odlyzko pointed out that while European countries had succeeded in introducing similar value coins (one UK Pound, ten French Francs, one Euro and so fourth) the migration had been forced by the withdrawal of the paper equivalent. Since such coercion was never going to take place for payments on the Internet he suggested that user inertia would always inhibit the growth Internet cash systems.

Odlyzko's second point was that vendors would prefer to sell bundles of goods an services rather than small quanta, because in general they can extract more money from the buyers that way. He illustrated the theory and practice of this with the example of Microsoft selling the full suite of Office products for much less than the sum of the prices of the individual components. The argument presented was that if vendors tend to sell in large blocks rather than small quantities then the need for MicroPayments is reduced.

The third thread of argument was that flat rate, rather than metered usage encourages greater use of a service, even when in practice the user may end up paying more. Odlyzko produced extensive evidence that users of telephone services much prefer flat rate charging for calls and metering reduced consumption even when the metered rate was cheaper. Given that flat rate pricing is preferred by the user the need for small payments is again diminished.

Finally, Odlyzko presented the idea that anonymous payment schemes were disliked by merchants because they got in the way of price discrimination, which is valued by vendors as a way of getting the most from their customers. There is a long history of differential pricing for different customers (or classes of customer) and he asserted that having anonymous payments meant that the merchant could not discriminate between the customers. Since MicroPayments tend to be anonymous, as a frequent way to reduce the transaction cost is to do away with much of the accounting and associated book keeping, such schemes inhibit price discrimination and are thus shunned by vendors.

Attendees were referred to an extensive selection of papers, reference material and other essays which are available on the Internet at:
`http://www.dtc.umn.edu/~odlyzko/`

3 Ron Rivest

Ron Rivest started by stating "Is there a need for micropayments – absolutely!"

He commented that while Odlyzko has compared MicroPayments against subscriptions in an either–or manner Rivest felt that they could coexist and each have a significant market share.

Defining MicroPayments as payments under $10, the domain where the processing cost is high relative to the value of the transaction, Rivest looked particularly at purchases of informational goods where the marginal cost of production is zero.

Rivest produced some statistics for the paid Internet content market for the first quarter of 2002. The market size for information downloads was said to to be about $300 million and growing at a rate of 100% a year. Of this content:

- 50% was sold as annual subscriptions
- 30% was sold as monthly subscriptions
- 14% was sold as single purchases
- 6% was sold as other subscriptions (e.g. 6-month)

The 14% of current paid internet content as pay-per-use was compared with (paper) newspapers, where about 31% is pay-per-use. It was pointed out that this implied that pay-per-use content on the Internet, with a suitable payment system, would likely take between 14% and 31% of the total sales, which in a $300M a quarter market is a fairly significant market share even if subscriptions continued as the dominant means of payment.

Rivest went on to mention the recent introduction by a mobile cellular phone provider of a '*69' service[1] on a subscription basis ($3/month); this was a total flop. When they re-priced it as 75¢/use, it sold beautifully. *"The surest way I know to kill a new service is to make it available* only *via subscription."* It was suggested that new users often like pay-per-use to try out new services; they may eventually become subscribers, but providing users with the ability to experiment cheaply with a new service can be a powerful way to acquire customers.

Rivest then quoted another recent study which estimated that the market for downloading articles was about $1.6 billion annually; much of this market is for single-use purchases.

Next Rivest raised the question "Is there a 'killer-app' for micropayments?" He suspected that there is, and that it is for music downloads. He asserted that the music industry is in big trouble; the incumbent players are struggling to find a new business model despite realising that their old ones are not working. They realise that they need to reprice, downwards, but don't know how to do that and have it make economic sense. A recent study by Forrester says that digital music downloads will generate $2 billion in new sales annually within five years, and that 39% of this new revenue will be from downloaded singles.

[1] Dialing *69 calls back the last party to call your phone line.

A related market study by Ipsos-Reid (Dec. 2002), is entitled, "U.S. Music Downloaders Prefer a Pay-Per-Download Transaction over Current Subscription-Based Offerings" was quoted. It found that:

- 28% of those over 12 years old have downloaded songs over the Internet (about 60 million people, and the figure is increasing)
- 31% of downloaders report having paid to download music; this indicates an increased willingness to pay for downloaded music.
- 27% of downloaders prefer a fee-based system, with 19% preferring a pay-per-download, while only 9% preferred a subscription system. This is a two-to-one vote in favor of pay-per-download.

Rivest also noted that the mobile ringtone market is now $1 billion per year and that this is entirely pay-per-download.

Moving on, Rivest noted that in that morning's[2] New York Times, there was a story entitled: "6 Retailers Plan Venture to Sell Music on the Web". The new venture is called "Echo", and the retailers include Best Buy, Tower Records, Virgin Entertainment, and three others. The article notes that CD sales have dropped from 785 million units in 2000 to 681 million units in 2002. The new venture may have pricing similar to Universal's current web site, at 99¢ per single download and $9.99 for an album. The article also noted that prices in the music industry also needs to come down.

From this Rivest concluded that Pay-per-use and pay-per-download will always be a significant part of the payment scene, especially in music. To actually make this happen, he asserted, one needs to keep transaction processing costs small. He noted that one current web site is selling singles at 99¢ each, but paying 35¢ to have each 99¢ payment processed by the credit-card company.

He went on to suggest that probabilistic payment customized to handle small amounts can help, citing his own work with Silvio Micali (which lead to the founding of Peppercoin) as well as work by Andrew Odlyzko and, in that morning's conference session, Levante. He noted that furthermore Moore's Law also helps, since one can now do an RSA digital signature faster than you can access the hard drive.

Raising the issue of "bearer-based systems", where possession of the bits is equivalent to possession of the value, Rivest lamented the absence of Financial Cryptography founder Robert Hettinga who had originally been scheduled to join the panel but noted that the Mondex system, developed under the auspices of Tim Jones, was also essentially a bearer system. Rivest stated that he had a fundamental problem with bearer-based system because of the ease with which bits, as opposed to atoms, can be copied. Therefore, he asserted, a bearer-based system can not exist without a global database to check for duplicates and in the end, it would be easier to have an account-based system, with smaller per-user databases of receipts and expenditures.

[2] The panel session took place on January 27th, 2003

4 Tim Jones

Tim Jones opened by stating that throughout history the vast majority of transactions had been based on the direct exchange of value for goods and services, rather than subscription based transactions. Why should the move to the electronic world fundamentally alter the payment choice that has been constant for hundreds of years?

He went on to say that often new technologies take in unexpected ways. For instance, the SMS[3] facility in GSM telephones was an afterthought to the original design but has not only gone on to be a major source of revenue for the phone companies has also spawned an entire youth culture. Jones illustrated this with an anecdote about driving along the motorway with his daughter and a friend of hers when they were passed by another car full of young men. When they men in the car saw Jones' daughter they pulled along side and held up a piece of paper with a cellular phone number on it, initiating a conversation by text message, rather than voice, which resulting in the girls going clubbing that evening with the occupants of the passing car.

Jones went on to discuss the way in which MicroPayments, and in particular peer-to-peer payment protocols, might be used to help the open source software community. He suggested there the current situation in which a great deal of good software was being developed by unpaid volunteers was not economically sound but that the range of payment options to reflect the value that people in this business world are creating is not adequate. This, he said, seems like a case where there are peers who appreciate value, and could assign it. If there were an effective way for lots of people to make small donations with ease then open source software authors would likely get great benefit from such a system, especially for popular software which gets used very widely. As an example pointed to the scenario where currently software is available in a "free" form and also in a "pay" form. If there were a viable mechanism to accept a payment of 50¢ then there might be no need for the free version but if millions of users each paid that amount then the author would be (extremely) well rewarded for their work. In the current system he stated that we are not exploring properly the price elesticity of demand.

Exploring the use of electronic cash in the physical world, rather than the online world, Jones stated that there were often cases where people were involved in low value transactions and it would be disappointing if electronic payments could not cope them. By way of an example he cited the case of the school bake sale where the purchaser is aged 7 and merchant ages 9 and historically their payment processing system consisted of a small box containing a few coins. He pointed out that the Mondex electronic cash system could cope with this scenario.

In general Jones was fairly optimistic about micropayments but he was less convinced that music would be the "killer app". He thought it was more likely that the killer application would end up surprising us all. In any case, it was

[3] Short Message Service

perhaps important to get a system up and running, so that these applications could be built and tried out. He thought that it was quite possible that the killer application would be open-source downloadable software, such as plug-ins. With the right price points (e.g. 50¢ per download), software creators could make a lot more money than they could by charging say $30 for a retail box with the software. Downloading a plug-in for 50¢ for say an Adobe plug-in is a "no-brainer" he asserted and so everyone who thought that they might need the software would be willing to pay, as long as the payment mechanism was there.

Finally, Jones said he wanted to enable almost anyone to become a merchant. He reiterated the example of children selling cookies for a bake sale, and wanted them to be able to do similar things on the Web.

5 Duncan Goldie-Scot

Duncan Goldie-Scot opened by saying that in his time as a journalist writing about electronic money schemes he had examined 29 systems and 28 of them failed. He noted how hard it is to get the business model right and said that most payment systems failed because they aimed for "global domination", which he felt just isn't on the cards. He thought that by aiming for smaller markets, one could succeed.

Goldie-Scot suggested that the use of payment systems is governed by transaction costs, by the economics of the process. As processing costs collapse so average transaction value can and will fall until we end up with MicroPayments. As these MicroPayments become possible so new markets and new payment models will emerge.

Referring to Andrew Odlyzko's argument for flat-rate pricing he stated that this is not incompatible with micropayments. He said that it is an historical anomaly that we pay our flat-rate internet accounts monthly. Accounts payable and accounts receivable departments only exist because batch processing makes economic sense. When it becomes economically viable to pay the flat-rate more frequently – weekly, daily or by the second – then it is to the cash-flow advantage of someone to do so. $20 a month for 100m AOL users adds up to $2bn in the wrong hands – either the consumer or AOL depending on whether it is paid in advance or arrears. That carries a real economic cost. If it were technically possible to pay for everything in real-time there would be huge gains in economic efficiency. But paying $0.0000077160 per second is not yet viable.

Goldie-Scot went on to raise the example of British Gas. British Gas sends out 120 million bills a year, quarterly in arrears. Assuming the average bill is for $300, some $36bn is tied up in the payment system. A common gas utility (Transco) manages pipes that are used by all the players in the market. British Gas pays the wholesale price of gas from the gas pool and charges its customers the retail price. Their only real business is managing customer accounts. Keeping track of its customers is a difficult business, as is chasing bad debts. These two elements, the cash flow cost and the cost of knowing and billing the customer, account for the greater part of each gas bill.

If British Gas could develop a model in which they were paid cash on delivery and if they didn't even have to know who their customers were, they could strip out a large part of their cost base. If one could TCP/IP enable gas meters that could be linked to bank accounts, token issuers or some online payment mechanism – pay as you go – then you would change the economics of the retail gas market. One could envisage cash on delivery for gas, electricity, water, telephony, music, video – indeed anything that streams.

6 Debate from the Floor

A number of interesting questions were raised from the floor and a lively debate ensued. While a full transcript of the debate is not available we highlight here some of the questions and answers.

In the context of Tim Jones' comments about desiring digital money schemes that work for the school bake sale as well as for larger scale transactions, it was asked what a ten year old might want to sell on-line. One answer was that she might well want to sell her own songs, to her friends. Another was that it was the size of transaction that mattered and that the Open Source community would find it fruitful to be able to accept transactions on that scale.

The issue of the real cost of credit card transactions was raised, with one attendee citing that he had had cause to pay a toll of 1,000 Italian Lira, about 55¢ at the time of writing, on an Italian motorway and that the toll booth had been willing to take a credit card for the transaction. It seemed unlikely that the Italian government were paying a 25¢ overhead for that transaction. On the other hand another delegate pointed out that the State of California charges a $4 surcharge on all credit card transactions, irrespective of size, as it is supposedly easier than working out the real cost. It was generally agreed that much of the charge to the merchant was related to the cost of fraud; in places where there was likely to be less fraud the charges are lower. It was also commented that a large part of the cost is credit risk and thus transaction charges on debit cards are lower than on credit cards.

In the context of charging of phone calls it was pointed out that in Europe the use of pre-paid mobile phone services had overtaken the use of subscription services. Andrew Odlyzko said that this had resulted in a fall in the revenue per subscriber and suggested that this was evidence for his model but Tim Jones countered that this was in fact simply indication that the market was maturing; the revenue per subscriber for users in each class off subscriber had remained fairly stable but the mobile phone companies were now extracting money from new, less well off, users and while the total revenue was going up the effect of these new users was necessarily to bring the average down.

The next topic to come up was to do with the attention span of users and their willingness to make conscious decisions about payment. Andrew Odlyzko pointed to research from his time at AT&T in the mid seventies when metered local rate calls were tried. They were unpopular, even with the people who benefited from lower call charges as a result, and one of the reasons cited was to

do with the user not wanting to have to make a decision about cost. Ron Rivest stated that MicroPayment schemes would need to be very easy to use to become popular and pointed to work by Dan Ariely at the MIT Media Lad regarding the handling of this. Tim Jones suggested that this was really all part of the more general problem of user acceptance of the payment technology.

Ron Rivest questioned Andrew Odlyzko's assertion that price discrimination was impossible with MicroPayments. Odlyzko replied that it was not impossible but that anonymity made it harder. It was pointed out from the floor that the tacking system used by merchants such as Amazon were more or less orthogonal to the payment processing side.

7 Conclusions

Despite the wide variety of positions represented during the course of the discussion the general consensus seemed to be that MicroPayments do have some roll to play in the future of digital money. It is far from clear what form they will take but there is clearly a need for a simple, easy to use payment system for small value transactions which will not consume too large a fraction of the transaction value in processing charges. While such a system is unlikely to dominate the on line payments space the total value of even a modest fraction of all payments represents a huge amount of money.

In short, yes, we probably do need MicroPayments.

The Case Against Micropayments

Andrew Odlyzko

Digital Technology Center, University of Minnesota,
499 Walter Library, 117 Pleasant St. SE,
Minneapolis, MN 55455, USA
odlyzko@umn.edu
http://www.dtc.umn.edu/∼odlyzko

Abstract. Micropayments are likely to continue disappointing their advocates. They are an interesting technology. However, there are many non-technological reasons why they will take far longer than is generally expected to be widely used, and most probably will play only a minor role in the economy.

1 Introduction

This is an extended version of my remarks at the panel on "Does anyone really need MicroPayments?" at the Financial Cryptography 2003 Conference. For a report on the entire panel, see [20].

Micropayments are the technology of the future, and always will be. This was said about gallium arsenide (GaAs) over a decade ago, and has proven to be largely accurate. Although GaAs has found some niche applications (especially in high frequency communications), silicon continues to dominate the semiconductor industry.

The fate of micropayments is likely to be similar to that of gallium arsenide. They may become widespread eventually, but only after a long incubation period. They are also likely to play only a minor role in the economy. The reasons differ from the ones for the disappointments with GaAs. GaAs is playing a minor role because of technology trends. Silicon has improved faster than had been expected, and GaAs more slowly. On the other hand, the obstacles to micropayment adoption have very little to do with technology, and are rooted in economics, sociology, and psychology. Known micropayment schemes appear more than adequate in terms of providing low cost operations and adequate security. What is missing are convincing business cases.

This note is not a general survey of micropayments (see [1] for that, for example). It does not even present full details of the arguments against micropayments. Instead, it summarizes what appear to be the main obstacles to micropayment adoption. References are primarily to my own papers that are relevant, and those papers contain more detailed arguments and references. (Many of the arguments cited here have also been made by others, for example Clay Shirky [18].)

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 77–83, 2003.
© Springer-Verlag Berlin Heidelberg 2003

There have been and continue to be many proponents of micropayments. For example, Bob Metcalfe was an ardent advocate for them while he was a columnist for InfoWorld, arguing they were indispensable for a healthy Internet, and continues to believe they are inevitable. More recently, Merrill Lynch's Technology Strategist, Steve Milunovich, has also endorsed micropayments as a promising technology [8]. Many people have proposed to solve the spam problem by requiring micropayments on email (to be paid to the service providers or to recipients), to raise costs to spammers. The potential of micropayment appears high enough that even though many micropayment startups have folded, new ones keep springing up.

While I am pessimistic about micropayments, I am not opposed to them. The standard arguments for micropayments do have some validity. I have worked on several schemes, and together with S. Jarecki coinvented the probabilistic polling scheme [7]. However, while that work was being done during the summer of 1996, I was also involved in another study, of the economics of ecommerce. The research of that study led to a paper that predicted explicitly that micropayments were destined for only a marginal role in the economy [3]. Since that time, I have accumulated a variety of additional arguments supporting the pessimistic conclusion of [3].

As usual, micropayments in this note refer to systems where value changes hands at the time of the transaction. Accounted systems, such as electricity meters, which keep track of tiny transactions and bill for them at the end of a period, are not micropayments in this sense. Thus the arguments against micropayments here do not rule out microtransactions such as purchases of ring tones from cellular carriers or providers who bill through the cellular carriers. (However, some of the arguments do suggest that even such accounted systems are likely to be less important than various fixed fee subscription options.)

The following sections outline the main barriers to micropayment adoption. The final section discusses the most promising avenues for micropayment diffusion.

2 Competition from Other Payment Schemes

There is just one argument against micropayments that is based on technology. The same advances in computing and communications that make implementations of micropayment schemes feasible are also enabling competing payment systems (especially credit and debit cards) to economically handle decreasingly small transactions. Hence the market for handling small transactions that only micropayments can handle is shrinking rapidly. (Note that this is similar to what happened in semiconductors. There improvements in silicon technologies have limited the areas that seemed likely to be taken over by gallium arsenide.)

The slow pace of change in payment systems (discussed in the next section) strengthens this argument significantly.

3 Payment Evolution on Non-Internet Time

Probably the most damaging myth behind the high-tech bubble of the late 1990s was that of "Internet time," that technology and the economy were changing far faster than before. While there are a few small grains of truth to this, overall the pace of change has not accelerated all that much. In particular, new technologies still take on the order of a decade to diffuse widely [11], [15]. Changes in payment systems tend to be even slower [12]. (In fact, international comparisons of payment systems provide interesting examples for discussions of "path dependence," "lock-in," and similar concepts.) As a simple example, consider credit cards. They are ubiquitous in North America and many other industrialized countries. They are even spreading in countries like Germany, where it had been claimed for a long time that there would be no room for them for institutional and cultural factors. However, it took credit cards several decades to achieve their high penetration [2].

As yet another example, debit cards (which were common in other countries for a long time) have only recently achieved significant penetration in the United States. The reason for their adoption is largely the push by banks, which found this to be a high-profit opportunity. Thus banks played the roles of "forcing agents" discussed in [11] that can sometimes propel faster adoption of new technologies than would have happened otherwise. Even so, the progress of debit transactions in the United States has not been very rapid. When there are no "forcing agents," progress is often glacial, as in the lack of acceptance of the Sacagawea dollar coin. It was introduced several years ago without the serious design flaws of the earlier Susan B. Anthony coin, but is practically never used in early 2003. (By contrast, other countries, such as Britain, France, Germany, or Japan, that did successfully introduce large denomination coins, did it by government fiat, by withdrawing corresponding bills from circulation.)

The slow pace of adoption of new payment schemes does not doom micropayments. However, it does demolish the hopes of venture capitalists who invest in micropayment startups, and certainly goes counter to the general expectations of micropayment proponents for rapid acceptance. (In particular, it does decrease the "first mover advantage" that many startups count on.) It also leaves an opening for competing payment systems to take over much of those parts of the economy that seemed natural preserves for micropayments, as is discussed in the preceding section.

4 Bundling

Proponents of micropayments have claimed that they would open up new avenues for commerce. They would enable microtransactions, such as newspapers selling individual stories instead of entire issues, Web sites selling access to individual pages, and even ISPs charging for each packet transmitted. We have seen very little of that, and for good reasons. In general, it is to the sellers' advantage to sell bundles of goods, as that maximizes their profits. As an example, the

Microsoft Office bundle typically sells for about half of the sum of the prices of components (Word, PowerPoint, ...). This is not done out of charitable impulses, but to increase revenues and profits. What Microsoft and other sellers are doing is taking advantage of uneven preferences among their customers for different parts of the bundle.

The advantages of bundling in increasing revenues have been known in economics for about four decades. There are various mathematical models that demonstrate how useful bundling is, and how its advantages depend on number of items in the bundle, distribution of customer preferences, marginal costs, and other factors. (For some references, see [3].) The general conclusion is that aggregation strategies tend to be more profitable for sellers. This argument again does not doom micropayments, since it is well known that mixed strategies (offering both bundles and individual items, but with prices of separate items higher than they would be if bundling were not feasible) are usually more profitable than pure bundling. (And indeed Microsoft does sell Word by itself.) However, this again limits the range of transactions that seemed the natural domain for micropayments.

5 Resistance to Anonymity

Micropayments have often been promoted as providing the anonymity of cash transactions. However, while anonymity is often desired by consumers, it is resisted by both governments and sellers. Government resistance is based on concerns about money laundering, tax evasion, terrorism funding, and other illegal activities, and is well understood. Commercial entities, on the other hand, might be expected to be more receptive to their customers' wishes. In practice, though, they are the ones most responsible for the persistent privacy erosion we see. The reason is that sellers have strong incentives to price discriminate, either explicitly or implicitly, through versioning and other techniques. Therefore they have strong interests in avoiding anonymous transactions [10], [16]. Thus another factor that has been widely hailed as an advantage of micropayments works against them.

6 Behavioral Economics

Behavioral economics, the study of what had for a long time been dismissed as the economicly irrational behavior of people, is finally becoming respectable within economics. In marketing, it has long been used in implicit ways. One of the most relevant findings for micropayments is that consumers are willing to pay more for flat-rate plans than for metered ones. This appears to have been discovered first about a century ago, in pricing of local telephone calls [13], but was then forgotten. It was rediscovered in the 1970s in some large scale experiments done by the Bell System [3]. There is now far more evidence of this, see references in [13], [14]. As one example of this phenomenon, in the fall

of 1996, AOL was forced to switch to flat rate pricing for Internet access. The reasons are described in [19]:

> What was the biggest complaint of AOL users? Not the widely mocked and irritating blue bar that appeared when members downloaded information. Not the frequent unsolicited junk e-mail. Not dropped connections. Their overwhelming gripe: the ticking clock. Users didn't want to pay by the hour anymore. ... Case had heard from one AOL member who insisted that she was being cheated by AOL's hourly rate pricing. When he checked her average monthly usage, he found that she would be paying AOL more under the flat-rate price of $19.95. When Case informed the user of that fact, her reaction was immediate. 'I don't care,' she told an incredulous Case. 'I am being cheated by you.'

The lesson of behavioral economics is thus that small payments are to be avoided, since consumers are likely to pay more for flat-rate plans. This again argues against micropayments.

7 Incentives to Increase Usage

Both behavioral economics and conventional economic utility analysis argue that in an environment of low marginal costs (which are increasingly prevalent in our economy), sellers have a strong incentive to increase usage of their goods and services. Although "network effects" were a much-overused mantra of the dot-com bubble, they are real. As one example, Bill Gates said in 1998 [17]:

> Although about three million computers get sold every year in China, people don't pay for the software. Someday they will, though. And as long as they're going to steal it, we want them to steal ours. They'll get sort of addicted, and then we'll somehow figure out how to collect sometime in the next decade.

Any kind of barrier to usage, such as explicit payment, serves to discourage usage. (That was the basis for the prediction in [9] that pay-per-view was doomed in scholarly publishing.) Even small barriers, such as having to pay for for individual pages, act as a severe deterrent to usage. During the mid- to late-1990s, several scholarly publishers experimented with a variety of payment schemes for science, technology, and medical information through the PEAK system. The conclusion that the main publisher in the experiment, Elsevier, drew, was very clear [6]:

> [Elsevier's] goal is to give people access to as much information as possible on a flat fee, unlimited use basis. [Elsevier's] experience has been that as soon as the usage is metered on a per-article basis, there is an inhibition on use or a concern about exceeding some budget allocation.

The same arguments will be increasingly persuasive as we become more of an "attention economy" [5], in which the most scarce resource is human attention. The incentives to increase usage argue for selling goods and services in ways that maximize usage, and nothing does that as well as flat-rate (or subscription) pricing. A general rule of thumb is that switching from metered to flat-rate pricing increases usage by 50 to 200 percent [13], [14]. As one particularly noteworthy example, when AOL switched to the unlimited usage plans in the fall of 1996, the average time spent online per subscriber tripled over the next year. Hence we should expect to see a continuing and even increasing dominance of flat-rate plans, and this again destroys much of the argument for micropayments.

As a final example, a recent story about new communication, information, and entertainment services stated that "[w]hat all these emerging services have in common is a business model based on subscriptions that are billed monthly or yearly" [4]. The sellers of these services are reacting to a variety of incentives mentioned in this and previous sections. While one can argue that widespread availability of micropayments might lead them to offer different payment options (for example, to encourage people to try out a novelty), this is unlikely, since accounted systems (with billing ultimately to a credit card, say) would be quite adequate for most of these services, had the sellers had real incentives to use them.

8 Conclusions

The general conclusion drawn from the discussion above is that there are many factors working against the success of micropayments. Even some of the features that seemed to be very attractive about micropayments, such as anonymity, work against them. The technologists have produced many micropayment schemes that are efficient and secure enough to be used widely. However, economics, sociology, and psychology place obstacles in the path of micropayments that are likely to keep them restricted to a marginal role in the economy forever.

Still, micropayments may become widespread. There are needs that micropayments are uniquely suited to fill. However, given all the obstacles that micropayments face, they are unlikely to succeed if offered as a service that requires special hardware or software. They are most likely to succeed if they piggyback on top of something that is already widely used, such as cell phones, or (in some places) mass-transit smart cards. When offered as an additional feature for something that is already carried by most of the population, micropayments might be able to overcome the usual chicken and egg problem, and find their (very likely small) niche in the economy.

References

1. Dingledine, R., Freedman, M.J., Molnar, D.: Accountability. In: Oram, A. (ed.): Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly, 2001. Available at ⟨http://www.freehaven.net/doc/oreilly/micropayments.txt⟩.

2. Evans, D., Schmalensee, R.: Paying with Plastic: The Digital Revolution in Buying and Borrowing. MIT Press, 1999.
3. Fishburn, P.C., Odlyzko, A.M., Siders, R.C.: Fixed Fee Versus Unit Pricing for Information Goods: Competition, Equilibria, and Price Wars, First Monday 2(7) (July 1997), ⟨http://firstmonday.org/⟩. Definitive version on pp. 167-189 in "Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property," B. Kahin and H. R. Varian, eds., MIT Press, 2000. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
4. Fixmer, R.: It Adds Up (and Up, and Up). New York Times, April 10, 2003.
5. Goldhaber, M.H.: The Attention Economy and The Net. First Monday **2** (no. 4) April 1997, ⟨http://firstmonday.org/⟩.
6. Hunter, K.: PEAK and Elsevier Science: Presented at the 2000 conference in Ann Arbor, Michigan, The Economics and Usage of Digital Library Collections. Available at ⟨http://www.si.umich.edu/PEAK-2000/program.htm⟩.
7. Jarecki, S., Odlyzko, A.M.: An Efficient Micropayment System Based on Probabilistic Polling. In: Hirschfeld, R. (ed.): Financial Cryptography. Lecture Notes in Computer Science, Vol. 1318. Springer-Verlag, (1997) 173–191. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
8. Milunovich, S.: Micropayment's Big Potential. Red Herring (Nov. 2002). Available at ⟨http://www.redherring.com/investor/2002/11/micropayments-110502.html⟩.
9. Odlyzko, A.M.: Tragic Loss or Good Riddance? The Impending Demise of Traditional Scholarly Journals. Intern. J. Human-Computer Studies **42** ((1995) 71–122. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
10. Odlyzko, A.M.: The Bumpy Road of Electronic Commerce. In: Maurer, H. (ed.): WebNet 96 – World Conf. Web Soc. Proc.. AACE (1996) 378–389. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
11. Odlyzko, A.M.: The Slow Evolution of Electronic Publishing. In: Meadows, A.J., Rowland, F. (eds.): Electronic Publishing '97: New Models and Opportunities. ICCC Press (1997) 4–18. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
12. Odlyzko, A.M.: The Future of Money. Unpublished 1998 manuscript, available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
13. Odlyzko, A.M.: The History of Communications and its Implications for the Internet. Unpublished 2000 manuscript, available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
14. Odlyzko, A.M.: Internet Pricing and the History of Communications. Computer Networks **36** (2001) 493–517. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
15. Odlyzko, A.M.: The Myth of Internet Time. Technology Review **104** (no. 3) (April 2001) 92–93. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
16. Odlyzko, A.M.: Privacy, Price Discrimination, and the Future of Ecommerce. Manuscript in preparation.
17. Schlender, B.: The Bill and Warren Show. Fortune, July 20, 1998.
18. Shirky, C.: The case against micropayments. O'Reilly Network Dec. 19, 2000. Available at ⟨http://www.oreillynet.com/pub/a/p2p/2000/12/19/micropayments.html⟩.
19. Swisher, K.: Aol.Com: How Steve Case Beat Bill Gates, Nailed the Netheads, and Made Millions in the War for the Web. Times Books (1998).
20. van Someren, N., Odlyzko, A., Rivest, R., Jones, T., Goldie-Scot, D.: Does anyone really need MicroPayments?: These proceedings.

On the Economics of Anonymity

Alessandro Acquisti[1], Roger Dingledine[2], and Paul Syverson[3]

[1] SIMS, UC Berkeley acquisti@sims.berkeley.edu
[2] The Free Haven Project arma@mit.edu
[3] Naval Research Lab syverson@itd.nrl.navy.mil

Abstract. Decentralized anonymity infrastructures are still not in wide use today. While there are technical barriers to a secure robust design, our lack of understanding of the incentives to participate in such systems remains a major roadblock. Here we explore some reasons why anonymity systems are particularly hard to deploy, enumerate the incentives to participate either as senders or also as nodes, and build a general model to describe the effects of these incentives. We then describe and justify some simplifying assumptions to make the model manageable, and compare optimal strategies for participants based on a variety of scenarios.

Keywords: Anonymity, economics, incentives, decentralized, reputation

1 Introduction

Individuals and organizations need anonymity on the Internet. People want to surf the Web, purchase online, and send email without exposing to others their identities, interests, and activities. Corporate and military organizations must communicate with other organizations without revealing the existence of such communications to competitors and enemies. Firewalls, VPNs, and encryption cannot provide this protection; indeed, Diffie and Landau have noted that traffic analysis is the backbone of communications intelligence, not cryptanalysis [9].

With so many potential users, it might seem that there is a ready market for anonymity services — that is, it should be possible to offer such services and develop a paying customer base. However, with one notable exception (the Anonymizer [2]) commercial offerings in this area have not met with sustained success. We could attribute these failures to market immaturity, and to the current economic climate in general. However, this is not the whole story.

In this paper we explore the incentives of participants to offer and use anonymity services. We set a foundation for understanding and clarifying our speculations about the influences and interactions of these incentives. Ultimately we aim to learn how to align incentives to create an economically workable system for users and infrastructure operators.

Section 2 gives an overview of the ideas behind our model. Section 3 goes on to describe the variety of (often conflicting) incentives and to build a general model that incorporates many of them. In Section 4 we give some simplifying

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 84–102, 2003.
© Springer-Verlag Berlin Heidelberg 2003

assumptions and draw conclusions about certain scenarios. Sections 5 and 6 describe some alternate approaches to incentives, and problems we encounter in designing and deploying strong anonymity systems.

2 The Economics of Anonymity

Single-hop web proxies like the Anonymizer protect end users from simple threats like profile-creating websites. On the other hand, users of such commercial proxies are forced to trust them to protect traffic information. Many users, particularly large organizations, are rightly hesitant to use an anonymity infrastructure they do not control. However, on an open network such as the Internet, running one's own system won't work: a system that carries traffic for only one organization will not hide the traffic entering and leaving that organization. Nodes must carry traffic from others to provide cover. The only viable solution is to distribute trust. That is, each party can choose to run a node in a shared infrastructure, if its incentives are large enough to support the associated costs. Users with more modest budgets or shorter-term interest in the system also benefit from this decentralized model, because they can be confident that a few colluding nodes are unlikely to uncover their anonymity.

Today, however, few people or organizations are willing to run these nodes. In addition to the complexities of configuring current anonymity software, running a node costs a significant amount of bandwidth and processing power, most of which is used by 'freeloading' users who do not themselves run nodes. Moreover, when administrators are faced with abuse complaints concerning illegal or antisocial use of their systems, the very anonymity that they're providing precludes the usual solution of suspending users or otherwise holding them accountable.

Unlike confidentiality (encryption), anonymity cannot be created by the sender or receiver. Alice cannot decide by herself to send anonymous messages — she must trust the infrastructure to provide protection, and others must use the same infrastructure. Anonymity systems use messages to hide messages: senders are consumers of anonymity and also providers of the cover traffic that creates anonymity for others. Thus users are better off on crowded systems because of the noise other users provide.

Because high traffic is necessary for strong anonymity, agents must balance their incentives to find a common equilibrium, rather than each using a system of their own. The high traffic they create together also enables better performance: a system that processes only light traffic must delay messages to achieve adequately large anonymity sets. But systems that process the most traffic do not necessarily provide the best hiding: if trust is not well distributed, a high volume system is vulnerable to insiders and attackers who target the trust bottlenecks.

Anonymity systems face a surprisingly wide variety of direct anonymity-breaking attacks [3,20]. Additionally, adversaries can also attack the efficiency or reliability of nodes, or try to increase the cost of running nodes. All of these factors combine to threaten the *anonymity* of the system. As Back et al. point out, "in anonymity systems usability, efficiency, reliability and cost become *secu-*

rity objectives because they affect the size of the user base which in turn affects the degree of anonymity it is possible to achieve." [3]

We must balance all of these tradeoffs while we examine the incentives for users and node operators to participate in the system.

3 Analytic Framework

In this section and those that follow, we formalize the economic analysis of why people might choose to send messages through mix-nets.[1] We discuss the incentives for agents to participate either as senders or also as nodes, and we propose a general framework to analyze these incentives. In the next section we consider various applications of our framework, and then in Section 5 we examine alternate incentive mechanisms.

We begin with two assumptions: the agents want to send messages to other parties, and the agents value their anonymity. How various agents might value their anonymity will be discussed below.

An agent i (where $i = (1, ..., n)$ and n is the number of potential participants in the mix-net) bases her strategy on the following possible actions a_i:

1. Act as a user of the system, specifically by sending (and receiving) her own traffic over the system, a_i^s, and/or agreeing to receive dummy traffic through the system, a_i^r. (*Dummy traffic* is traffic whose only purpose is to obscure actual traffic patterns.)
2. Act as an honest node, a_i^h, by receiving and forwarding traffic (and possibly acting as an exit node), keeping messages secret, and possibly creating dummy traffic.
3. Act as a dishonest node, a_i^d, by pretending to forward traffic but not doing so, by pretending to create dummy traffic but not doing so (or sending dummy traffic easily recognizable as such), or by eavesdropping traffic to compromise the anonymity of the system.
4. Send messages through conventional non-anonymous channels, a_i^n, or send no messages at all.

Various benefits and costs are associated with each agent's action and the simultaneous actions of the other agents. The expected benefits include:

1. Expected benefits from sending messages anonymously. We model them as a function of the subjective value each agent i places on the information successfully arriving at its destination, v_{r_i}; the subjective value of keeping her identity anonymous, v_{a_i}; the perceived level of anonymity in the system, p_{a_i} (the subjective probability that the sender and message will remain anonymous); and the perceived level of reliability in the system, p_{r_i} (the subjective probability that the message will be delivered). The subjective value

[1] Mixes were introduced by David Chaum (see [6]). A mix takes in a batch of messages, changes their appearance, and sends them out in a new order, thus obscuring the relation of incoming to outgoing messages.

of maintaining anonymity could be related to the profits the agent expects to make by keeping that information anonymous, or the losses the agents expects to avoid by keeping that information anonymous. We represent the level of anonymity in the system as a function of the traffic (number of agents sending messages in the system, n_s), the number of nodes (number of agents acting as honest nodes, n_h, and as dishonest nodes, n_d), and the decisions of the agent. We assume the existence of a function that maps these factors into a probability measure $p \in [0, 1]$.[2] In particular:

- The level of anonymity of the system is positively correlated to the number of users of the system.
- Acting as an honest node improves anonymity. Senders who do not run a node may accidentally choose a dishonest node as their first hop, significantly decreasing their anonymity (especially in low-latency anonymity systems where end-to-end timing attacks are very hard to prevent [3]). Further, agents who run a node can undetectably blend their message into their node's traffic, so an observer cannot know when the message is sent.
- The relation between the number of nodes and the probability of remaining anonymous might not be monotonic. For a given amount of traffic, sensitive agents might want fewer nodes in order to maintain large anonymity sets. But if some nodes are dishonest, users may prefer more honest nodes (to increase the chance that messages go through honest nodes). Agents that act as nodes may prefer fewer nodes, to maintain larger anonymity sets at their particular node. Hence the probability of remaining anonymous is inversely related to the number of nodes but positively related to the ratio of honest/dishonest nodes. (On the other hand, improving anonymity by reducing the number of nodes can be taken too far — a system with only one node may be easier to monitor and attack. See Section 5 for more discussion.)

If we assume that honest nodes always deliver messages that go through them, the level of reliability in the system is then an inverse function of the share of dishonest nodes in the system, n_d/n_h.

2. Benefits of acting as a node (nodes might be rewarded for forwarding traffic or for creating dummy traffic), b_h.
3. Benefits of acting as a dishonest node (from disrupting service or by using the information that passes through them), b_d.

The possible expected costs include:

1. Costs of sending messages through the anonymous system, c_s, or through a non-anonymous system, c_n. These costs can include both direct financial

[2] Information theoretic anonymity metrics [8,22] probably provide better measures of anonymity: such work shows how the level of anonymity achieved by an agent in a mix-net system is associated to the particular structure of the system. But probabilities are more tractable in our analysis, as well as better than the common "anonymity set" representation.

costs such as usage fees, as well as implicit costs such as the time to build and deliver messages, learning curve to get familiar with the system, and delays incurred when using the system. At first these delays through the anonymous system seem positively correlated to the traffic n_s and negatively correlated to the number of nodes n_h. But counterintuitively, more messages per node might instead *decrease* latency because nodes can process batches more often; see Section 5. In addition, when message delivery is guaranteed, a node might always choose a longer route to reduce risk. We could assign a higher c_s to longer routes to reflect the cost of additional delay. We also include here the cost of receiving dummy traffic, c_r.

2. Costs of acting as an honest node, c_h, by receiving and forwarding traffic, creating dummy traffic, or being an exit node (which involves potential exposure to liability from abuses). These costs can be variable or fixed. The fixed costs, for example, are related to the investments necessary to setup the software. The variable costs are often more significant, and are dominated by the costs of traffic passing through the node.
3. Costs of acting as dishonest node, c_d (again carrying traffic; and being exposed as a dishonest node may carry a monetary penalty).

In addition to the above costs and benefits, there are also *reputation* costs and benefits from: being observed to send or receive anonymous messages, being perceived to act as a reliable node, and being thought to act as a dishonest node.

Some of these reputation costs and benefits could be modelled endogenously (e.g., being perceived as an honest node brings that node more traffic, and therefore more possibilities to hide that node's messages; similarly, being perceived as a dishonest node might bring traffic away from that node). In this case, they would enter the payoff functions only indirectly through other parameters (such as the probability of remaining anonymous) and the changes they provoke in the behavior of the agents. In other cases, reputation costs and benefits might be valued *per se*. While we do not consider either of these options in the simplified model below, Sections 5 and 6 discuss the impact of reputation on the model.

We assume that agents want to maximize their expected payoff, which is a function of expected benefits minus expected costs. Let S_i denote the set of strategies available to agent i, and s_i a certain member of that set. Each strategy s_i is based on the the actions a_i discussed above. The combination of strategies $(s_1, ..., s_n)$, one for each agent who participates in the system, determines the outcome of a game as well as the associated payoff for each agent. Hence, for each complete strategy profile $s = (s_1, ..., s_n)$ each agent receives the expected payoff $u_i(s)$ through the payoff function $u(.)$. We represent the payoff function for each agent i in the following form:

$$u_i = u \begin{pmatrix} \theta\left[\gamma\left(v_{r_i}, p_{r_i}\left(n_h, n_d, a_h^s\right)\right), \partial\left(v_{a_i}, p_{a_i}\left(n_s, n_h, n_d, a_h^s\right)\right), a_i^s\right] + b_h a_i^h + b_d a_i^d \\ -c_s\left(n_s, n_h\right) a_i^s - c_h\left(n_s, n_h, n_d\right) a_i^h - c_d\left(..\right) a_i^d - c_r\left(..\right) a_i^r + (b_n - c_n) a_i^n \end{pmatrix}$$

where $\theta(.), \gamma(.)$, and $\partial(.)$ are unspecified functional forms. The payoff function $u(.)$ includes the costs and benefits for all the possible actions of the agents,

including *not* using the mix-net and instead sending the messages through a non-anonymous channel. We can represent the various strategies by using dummy variables for the various a_i.[3] We note that the probabilities of a message being delivered and a message remaining anonymous are weighted with the values v_{r_i}, v_{a_i}, respectively. This is because different agents might value anonymity and reliability differently, and because in different scenarios anonymity and reliability for the same agent might have different impacts on her payoff.

In Section 4, we will make a number of assumptions that will allow us to simplify this equation and model certain scenarios. We present here for the reader's convenience a table summarizing those variables that will appear in both the complete and simplified equations, as well as one that describes the variables used only in the more complete equation above.

Variables used in both full and simple payoff equations	
	u_i payoff for agent i
	v_{a_i} disutility i attaches to message exposure
	p_a simple case: $p_{a_i} = p_a$ for all i. See next table.
number of nodes (other than i) in mix-net	n_s sending agents (sending nodes)
	n_h honest nodes
	n_d dishonest nodes
dummy variables: 1 if true, 0 otherwise	a_i^h i is an honest node and sending agent
	a_i^s i sends through the mix-net
costs	c_h of running an honest node
	c_s of sending a message through the mix-net

Variables used only in full payoff equation	
	v_{r_i} value i attaches to sent message being received
	p_{a_i} prob. for i that a sent message loses anonymity
	p_r prob. that message sent through mix-net is received
benefits	b_h of running an honest node
	b_d of running a dishonest node
	b_n of sending a message around the mix-net
dummy variables	a_i^d i runs a dishonest node
	a_i^n i sends message around the mix-net
	a_i^r i receives dummy traffic
costs	c_d of running a dishonest node
	c_r of receiving dummy traffic
	c_n of sending a message around the mix-net

Note also that the costs and benefits from sending the message could be distinct from the costs and benefits from keeping the *information* anonymous. For example, when Alice anonymously purchases a book, she gains a profit equal

[3] For example, if the agent chooses not to send the message anonymously, the probability of remaining anonymous p_{a_i} will be equal to zero, $a^{s,d,r,h}$ will be zero too, and the only cost in the function will be c_n.

to the difference between her valuation of the book and its price. But if her anonymity is compromised during the process, she could incur losses (or miss profits) completely independent from the price of the book or her valuation of it. The payoff function $u(.)$ above allows us to represent the duality implicit in all privacy issues, as well as the distinction between the value of sending a message and the value of keeping it anonymous:

Anonymity	*Reliability*
Benefit from remaining anonymous / cost avoided by remaining anonymous, or	Benefit in sending message that will be received / cost avoided by sending such a message, or
Cost from losing anonymity / profits missed because of loss of anonymity	Cost from a message not being received / profits missed by message not being received

Henceforth, we will consider the direct benefits or losses rather than their dual opportunity costs or avoided costs. Nevertheless, the above representation allows us to formalize the various possible combinations. For example, if a certain message is sent to gain some benefit, but anonymity must be protected in order to avoid losses, then v_{r_i} will be positive while v_{a_i} will be negative and p_{a_i} will enter the payoff function as $(1 - p_{a_i})$. On the other side, if the agent must send a certain message to avoid some losses but anonymity ensures her some benefits, then v_{r_i} will be negative and p_{r_i} will enter the payoff function as $(1 - p_{r_i})$, while v_{a_i} will be positive.[4]

With this framework we can compare, for example, the losses due to compromised anonymity to the costs of protecting it. An agent will decide to protect herself by spending a certain amount if the amount spent in defense plus the expected losses for losing anonymity after the investment are less than the expected losses from not sending the message at all.

4 Applying the Model

In this section we apply the above framework to simple scenarios. We make a number of assumptions to let us model the behavior of mix-net participants as players in a repeated-game, simultaneous-move game-theoretic framework. Thus we can analyze the economic justifications for the various choices of the participants, and compare design approaches to mix-net systems.

Consider a set of n_s agents interested in sending anonymous communications. Imagine that there is only one system which can be used to send anonymous messages, and one other system to send non-anonymous messages. Each agent has three options: only send her own messages through the mix-net; send her messages but also act as a node forwarding messages from other users; or don't use the system at all (by sending a message without anonymity, or by not sending

[4] Being certain of staying anonymous would therefore eliminate the risk of v_{a_i}, while being certain of losing anonymity would impose on the agent the full cost v_{a_i}. Similarly, guaranteed delivery will eliminate the risk of losing v_{r_i}, while delivery failure will impose the full cost v_{r_i}.

the message). Thus initially we do not consider the strategy of choosing to be a bad node, or additional honest strategies like creating and receiving dummy traffic.

We represent the game as a simultaneous-move, repeated game because of the large number of participants and because of the impact of earlier actions on future strategies. A large group will have no discernable or agreeable order for the actions of all participants, so actions can be considered simultaneous. The limited commitment produced by earlier actions allows us to consider a repeated-game scenario.[5] These two considerations suggest against using a sequential approach of the Stackelberg type [14, Ch. 3]. For similar reasons we also avoid a "war of attrition/bargaining model" framework (see for example [21]) where the relative impatience of players plays an important role.

4.1 Adversary

Although strategic agents cannot choose to be bad nodes in this simplified scenario, we still assume there is a percentage of bad nodes and that agents respond to this possibility. Specifically we assume a global passive adversary (GPA) that can observe all traffic on all links (between users and nodes, between nodes, and between nodes or users and recipients). Additionally, we also study the case when the adversary includes some percentage of mix nodes. In choosing strategies agents will attach a subjective probability to arbitrary nodes being compromised — all nodes not run by the agent are assigned the same probability of being compromised. This factor influences their assessment of the anonymity of messages they send. A purely passive adversary is unrealistic in most settings, e.g., it assumes that hostile users never selectively send messages at certain times or over certain routes, and nodes and links never selectively trickle or flood messages [23]. Nonetheless, a *global* passive adversary is still quite strong, and thus a typical starting point of anonymity analyses.

4.2 Honest Agents

If a user only sends messages, the cost of using the anonymous service is c_s. This cost might be higher than using the non-anonymous channel, c_n, because of usage fees, usage hassles, or delays. To keep things simple, we assume that all messages pass through the mix-net in fixed-length free routes, so that we can write c_s as a fixed value, the same for all agents. Users send messages at the same time, and only one message at a time. We also assume that routes are chosen randomly by users, so that traffic is uniformly distributed among the nodes.[6]

If a user decides to be a node, her costs increase with the volume of traffic (we focus here on the traffic-based variable costs). We also assume that all agents know the number of agents using the system and which of them are acting as

[5] In Section 3 we have highlighted that, for both nodes and simpler users, variable costs are more significant than fixed costs.

[6] Reputation considerations might alter this point; see Section 5.

nodes. We also assume that all agents perceive the same level of anonymity in the system based on traffic and number of nodes, hence $p_{a_i} = p_a$ for all i. Finally, we imagine that agents use the system because they want to avoid potential losses from not being anonymous. This subjective sensitivity to anonymity is represented by v_{a_i} (we can initially imagine v_{a_i} as a continuous variable with a certain distribution across all agents; see below). In other words, we initially focus on the goal of remaining anonymous given an adversary that can control some nodes and observe all communications. Other than anonymity, we do not consider any potential benefit or cost, e.g., possible greater reliability, from sending around the mix-net. We later comment on the additional reliability issues.

$$u_i = -v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d, a_i^h\right)\right) - c_s a_i^s - c_h\left(n_s, n_h, n_d\right) a_i^h - v_{a_i} a_i^n$$

Thus each agent i tries to *minimize* the costs of sending messages and the risk of being tracked. The first component is the probability that anonymity will be lost given the number of agents sending messages, the number of them acting as honest and dishonest nodes, and the action a of agent i itself. This chance is weighted by v_{a_i}, the disutility agent i derives from its message being exposed. We also include the costs of sending a message through the mix-net, acting as a node when there are n_s agents sending messages over n_h and n_d nodes, and sending messages through a non-anonymous system, respectively. Each period, a rational agent can compare the payoff coming from each of these three one-period strategies.

Action	Payoff
a_s	$-v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d\right)\right) - c_s$
a_h	$-v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d, a_i^h\right)\right) - c_s - c_h\left(n_s, n_h, n_d\right)$
a_n	$-v_{a_i}$

We do not explicitly allow the agent to choose *not* to send a message at all, which would of course minimize the risk of anonymity compromise. Also, we do not explicitly report the value of sending a successful message. Both are simplifications that do not alter the rest of the analysis.[7]

While this model is simple, it allows us to highlight some of the dynamics that might take place in the decision process of agents willing to use a mix-net. We now consider various versions of this model.

[7] We could insert an action a^0 with a certain disutility or cost from not sending any message, and then solve the problem of minimizing the expected losses. Or, we could insert in the payoff function for actions $a^{s,h,n}$ also the payoff from successfully sending a message compared to not sending it (which could be interpreted also as an opportunity cost), and solve the dual problem of maximizing the expected payoff. Either way, the "exit" strategy for each agent will either be sending a message non-anonymously, or not sending it at all, depending on which option maximizes the expected benefits or minimizes the expected losses. Thereafter, we can simply compare the two other actions (being a user, or being also a node) to the optimal exit strategy.

Myopic Agents. Myopic agents do not consider the long-term consequences of their actions. They simply consider the status of the network and, depending on the payoffs of the one-period game, adopt a certain strategy. Suppose that a new agent with a privacy sensitivity v_{a_i} is considering using a mix-net with (*currently*) n_s users and n_h honest nodes.

Then if

$$\begin{gathered}-v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ <-v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s, \text{ and}\\ -v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ <-v_{a_i}\end{gathered}$$

agent i will choose to become a node in the mix-net. If

$$\begin{gathered}-v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ >-v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s, \text{ and}\\ -v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s<-v_{a_i}\end{gathered}$$

then agent i will choose to be a user of the mix-net. Otherwise, i will simply not use the mix-net.

Our goal is to highlight the economic rationale implicit in the above inequalities. In the first case agent i is comparing the benefits of the contribution to her own anonymity of acting as a node to the costs. Acting as a node dramatically increases anonymity, but it will also bring more traffic-related costs to the agent. Agents with high privacy sensitivity (high v_{a_i}) will be more likely to accept the trade-off and become nodes because they risk a lot by losing their anonymity, and because acting as nodes significantly increases their probabilities of remaining anonymous. On the other side, agents with a lower sensitivity to anonymity might decide that the costs or hassle of using the system are too high, and would not send the message (or would use non-anonymous channels).

Strategic Agents: Simple Case. Strategic agents take into consideration the fact that their actions will trigger responses from the other agents.

We start by considering only one-on-one interactions. First we present the case where each agent knows the other agent's type, but we then discuss what happens when there is uncertainty about the other agent's type.

Suppose that each of agent i and agent j considers the other agent's reaction function in her decision process. Then we can summarize the payoff matrix in the following way:[8]

[8] We use parameters to succinctly represent the following expected payoffs:

$$\begin{gathered}A_w=-v_w\left(1-p_a\left(n_s+2,n_h+2,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+2,n_h+2,n_d\right)\\ B_w=-v_w\left(1-p_a\left(n_s+2,n_h+1,n_d\right)\right)-c_s\\ C_w=-v_w\\ D_w=-v_w\left(1-p_a\left(n_s+2,n_h+1,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+2,n_h+1,n_d\right)\\ E_w=-v_w\left(1-p_a\left(n_s+1,n_h+1,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ F_w=-v_w\left(1-p_a\left(n_s+2,n_h,n_d\right)\right)-c_s\\ G_w=-v_w\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s\end{gathered}$$

Agent i / Agent j	a_j^h	a_j^s	a_j^n
a_i^h	A_i, A_j	D_i, B_j	E_i, C_j
a_i^s	B_i, D_j	F_i, F_j	G_i, C_j
a_i^n	C_i, E_j	C_i, G_j	C_i, C_j

As before, each agent has a trade-off between the cost of traffic and the benefit of traffic when being a node, and a trade-off between having more nodes and fewer nodes. In addition to the previous analysis, now the final outcome also depends on how much each agent knows about whether the other agent is honest, and how much she knows about the other agent's sensitivity to privacy.

Of course, for an explicit solution we need a specific functional form for the probability function.[9] Nevertheless, even at this abstract level of description this framework can be mapped into the model analyzed in [19] where two players decide simultaneously whether to contribute to a public good.

In our model, when for example $v_{a_i} \gg v_{a_j}$ and v_{a_i} is large, the disutility to player i from not using the system or not being a node will be so high that she will decide to be a node even if j might free ride on her. Hence if j values her anonymity, but not that much, the strategies a_i^h,a_j^s can be an equilibrium of the repeated game.

In fact, this model might have equilibria with free-riding even when the other agent's type is unknown. Imagine both agents know that the valuations v_{a_i}, v_{a_j} are drawn independently from a continuous, monotonic probability distribution. Again, when one agent cares about her privacy enough, and/or believes that there is a high probability that the opponent would act as a dishonest node, then the agent will be better off protecting her own interests by becoming a node (again see [19]). Of course the more interesting cases are those when these clear-cut scenarios do not arise, which we consider next.

Strategic Agents: Multi-player Case. Each player now considers the strategic decisions of a vast number of other players. Fudenberg and Levine [13] propose a model where each player plays a large set of identical players, each of which is "infinitesimal", i.e. its actions cannot affect the payoff of the first player. We define the payoff of each player as the average of his payoffs against the distribution of strategies played by the continuum of the other players. In other words, for each agent, we will have: $u_i = \sum_{n_s} u_i\,(a_i, a_{-i})$ where the notation represents the comparison between one specific agent i and all the others. Cooperative solutions with a finite horizon are often not sustainable when the actions of other agents are not observable because, by backward induction, each agent will have an incentive to deviate from the cooperative strategy. As compared to the analysis above with only two agents, now a defection of one agent might

[9] We have seen above, however, that privacy metrics like [8,22] do not directly translate into monotonic probability functions of the type traditionally used in game theory. Furthermore, the actual level of anonymity will depend on the mix-net protocol and topology (synchronous networks will provide larger anonymity sets than asynchronous networks for the same traffic divided among the nodes).

affect only infinitesimally the payoff of the other agents, so the agents might tend not to punish the defector. But then, more agents will tend to deviate and the cooperative equilibrium might collapse. "Defection", in fact, could be acting only as a user and refusing to be a node when the agent starts realizing that there is enough anonymity in the system and she no longer needs to be a node. But if too many agents act this way, the system might break down for lack of nodes, after which everybody would have to resort to non-anonymous channels.

We can consider this to be a "public good with free-riding" type of problem [7]. The novel point from a game-theoretic perspective is that the highly sensitive agents actually *want* some level of free-riding, to provide noise. On the other side, they do not want too much free-riding — for example from highly sensitive types pretending to be agents with low sensitivity — if it involves high traffic costs.

So, under which conditions will a system with many players not implode?

First, a trigger strategy might be agreed upon among the many agents, so that the deviation of one single player might be met by the reaction of all the others (as described in [13]). Of course the only punishment available here is making the system unavailable, which has a cost for all agents. In addition, coordination costs might be prohibitive. This is not a viable strategy.

Second, we must remember that highly sensitive agents, for a given amount of traffic, prefer to be nodes (because anonymity will increase) and prefer to work in systems with fewer nodes (else traffic gets too dispersed and the anonymity sets get too small). So, if v_{a_i} is particularly high, i.e. if the cost of not having anonymity is very high for the most sensitive agents, then they will decide to act as nodes regardless of what the others do. Also, if there are enough agents with lower v_{a_i}, again a "high" type might have an interest in acting alone if its costs of not having anonymity would be too high compared to the costs of handling the traffic of the less sensitive types.

In fact, when the valuations are continuously distributed, this *might* generate equilibria where the agents with the highest valuations v_{a_i} become nodes, and the others, starting with the "marginal" type (the agent indifferent between the benefits she would get from acting as node and the added costs of doing so) provide traffic.[10] This problem can be mapped to the solutions in [4] or [17]. At that point an equilibrium level of free-riding might be reached. This condition can be also compared to [15], where the paradox of informationally efficient markets is described.[11]

The problems start if we consider now a different situation. Rather than having a continuous distribution of valuations v_{a_i}, we consider two types of agents: the agent with a high valuation, $v_{a_i} = v_H$, and the agent with a low valuation, $v_{a_i} = v_L$. We assume that the v_L agents will simply participate sending traffic if the system is cheap enough for them to use (but see Section 6.3), and we also assume this will not pose any problem to the v_H type, which in fact has an

[10] Writing down specific equilibria, again, will first involve choosing appropriate anonymity metrics, which might be system-dependent.

[11] The equilibrium in [15] relies on the "marginal" agent who is indifferent between getting more information about the market and not getting it.

interest in having more traffic. Thus we can focus on the interaction between a subset of users: the identical high-types.

Here the "marginal" argument discussed above might not work, and coordination might be costly. In order to have a scenario where the system is self-sustaining and free, and the agents are of high and low types, the actions of the agents must be visible and the agents themselves must agree to react together to any deviation of a marginal player. In realistic scenarios, however, this will involve very high transaction/coordination costs, and will require an extreme (and possibly unlikely) level of rationality for the agents. This equilibrium will also tend to collapse when the benefits from being a node are not very high compared to the costs. Paradoxically, it also breaks down when an agent trusts another so much that she prefers to delegate away the task of being a node. The above considerations however also hint at other possible solutions to reduce coordination costs. We now consider some other mechanisms that can make these systems economically viable.

5 Alternate Incentive Mechanisms

As the self-organized system might collapse under some of the conditions examined above, we discuss now what economic incentives we can get from alternative mechanisms.

1. *Usage fee.* If participants pay to use the system, the "public good with free-riding" problem turns into a "clubs" scenario. The pricing mechanism must be related to how much the participants expect to use the system or how sensitive they are. Sensitive agents might support the others by offering them limited services for free, because they need their traffic as noise. The Anonymizer offers basic service at low costs to low-sensitivity agents (there is a cost in the delay, the limitation on destination addresses, and the hassle of using the free service), and offers better service for money. With usage fees, the cost of being a node is externalized. A hybrid solution involves distributed trusted nodes, supported through entry fees paid to a central authority and redistributed to the nodes. This was the approach of the Freedom Network from Zero-Knowledge Systems. The network was shut down because they were unable to sell enough clients to cover their costs.
2. *"Special" agents.* Such agents have a payoff function that considers the social value of having an anonymous system or are otherwise paid or supported to provide such service. If these agents are paid, the mechanism becomes similar to the hybrid solution discussed above, except anonymity-sensitive agents, rather than act as nodes, pass the money to a central authority. The central authority redistributes the funding among trusted entities acting as nodes.
3. *Public rankings and reputation.* A higher reputation not only attracts more cover traffic but is also a reward in itself. Just as the statistics pages for seti@home [5] encourage participation, publicly ranking generosity creates an incentive to participate. Although the incentives of public recognition and

public good don't fit in our model very well, we emphasize them because they explain most actual current node operators. As discussed above, reputation can enter the payoff function indirectly or directly (when agents value their reputation as a good itself).

If we publish a list of nodes ordered by safety (based on number of messages passing through the node), the high-sensitivity agents will gravitate to safe nodes, causing more traffic and improving their safety further (and lowering the safety of other nodes). In our model the system will stabilize with one or a few mix nodes. In reality, though, p_a is influenced not just by n_h but also by jurisdictional diversity — a given high-sensitivity sender is happier with a diverse set of mostly busy nodes than with a set of very busy nodes run in the same zone. Also, after some threshold of users, latency will begin to suffer, and the low sensitivity users will go elsewhere, taking away the nice anonymity sets.

More generally, a low-latency node may attract many low-sensitivity agents, and thus counterintuitively provide *better* anonymity than one that waits to batch many messages for greater security.

6 A Few More Roadblocks

6.1 Authentication in a Volunteer Economy

Our discussions so far indicate that it may in fact be plausible to build a strong anonymity infrastructure from a wide-spread group of independent nodes that each want good anonymity for their own purposes. In fact, the more jurisdictionally diverse this group of nodes, the more robust the overall system.

However, volunteers are problems: users don't know the node operators, and don't know whether they can trust them. We can structure system protocols to create better incentives for honest principals and to catch bad performance by others, e.g. by incorporating receipts and trusted witnesses [10], or using a self-regulating topology based on verifying reliability [11]. But even when this is feasible, identifying individuals is a problem. Classic authentication considers whether it's the right entity, but not whether the authenticated parties are distinct from one another. One person may create and control several distinct online identities. This *pseudospoofing* problem [12] is a nightmare when an anonymity infrastructure is scaled to a large, diffuse, peer-to-peer design; it remains one of the main open problems in the design of any decentralized anonymity service. The Advogato trust metric [16] and similar techniques rely on humans to make initial trust decisions, and then bound trust flow over a certification graph. However, so far none of these trust flow approaches have provided a clear solution to the problem. Another potential solution, a global PKI to ensure unique identities [24], is unlikely to emerge any time soon.

6.2 Dishonest Nodes vs. Lazy Nodes

We have primarily focused on the strategic motivations of honest agents, but the motivations of dishonest agents are at least as important. An anonymity-breaking adversary with an adequate budget would do best to provide very good service, possibly also attempting DoS against other high-quality providers. None of the usual metrics of performance and efficiency can identify dishonest nodes. Further, who calculates those metrics and how? If they depend on a centralized trusted authority, the advantages of diffusion are lost. Another approach to breaking anonymity is to simply attack the reliability or perceived reliability of the system — this attack flushes users to a weaker system just as military strikes against underground cables force the enemy to communicate over less secure channels.

On the other hand, when we consider strategic dishonest nodes we must also analyze their motivations as rational agents. A flat-out dishonest agent participates only to compromise anonymity or reliability. In doing so, however, a dishonest agent will have to consider the costs of reaching and maintaining a position from which those attacks are effective — which will probably involve gaining reputation and acting as a node for an extended period of time, a cost if the goal is to generally break reliability. Such adversaries will be in an arms race with protocol developers to stay undetected despite their attacks [11]. The benefits from successful attacks might be financial, as in the case of discovering and using sensitive information or a competitor's service being disrupted; or they could be purely related to personal satisfaction. The costs of being discovered as a dishonest node include rebuilding a new node's worth of reputation; but being noticed and exposed as the adversary may have very serious negative consequences for the attacker itself. (Imagine the public response if an Internet provider were found running dishonest nodes.) Thus, all things considered, it might be that the laws of economics work against the attacker as well.

A "lazy" node, on the other hand, wants to protect her own anonymity, but keeps her costs lower by not forwarding or accepting all of her incoming traffic. By doing so this node decreases the reliability of the system. While this strategy might be sounder than the one of the flat-out dishonest node, it also exposes again the lazy node to the risk of being recognized as a disruptor of the system. In addition, this tactic, by altering the flow of the traffic through her own node, might actually reduce the anonymity of that agent.

Surveys and analysis on actual attacks on actual systems (e.g., [18]) can help determine which forms of attacks are frequent, how dangerous they are, and whether economic incentives or technical answers are the best countermeasures.

6.3 Bootstrapping the System and Perceived Costs

Our models so far have considered the strategic choices of agents facing an already existing mix-net. We might even imagine that the system does not yet exist but that, before the first period of the repeated-game, all the players can

somehow know each other and coordinate to start with one of the cooperative equilibria discussed above.

But this does not sound like a realistic scenario. Hence we must discuss how a mix-net system with distributed trust can come to be. We face a paradox here: agents with high privacy sensitivity want lots of traffic in order to feel secure using the system. They need many participants with lower privacy sensitivities using the system first. The problem lies in the fact that there is no reason to believe the lower sensitivity types are more likely to be early adopters. In addition, their *perceived* costs of using the system might be higher than the real costs[12] — especially when the system is new and not well known — so in the strategic decision process they will decide against using the mix-net at all. Correct marketing seems critical to gaining critical mass in an anonymity system: in hindsight, perhaps Zero-Knowledge Systems would have gotten farther had it placed initial emphasis on usability rather than security.

Note that here again reliability becomes an issue, since we must consider both the benefits from sending a message *and* keeping it anonymous. If the benefits of sending a message are not that high to begin with, then a low sensitivity agent will have fewer incentives to spend anything on the message's anonymity. We can also extend the analysis from our model that considers the costs and benefits of a single system to the comparison of different systems with different costs/benefit characteristics. We comment more on this in the conclusion.

Difficulties in bootstrapping the system and the myopic behavior [1] of some users might make the additional incentive mechanisms discussed in Section 5 preferable to a market-only solution.

6.4 Customization and Preferential Service Are Risky Too

Leaving security decisions up to the user is traditionally a way to transfer cost or liability from the vendor to the customer; but in strong anonymity systems it may be unavoidable. For example, the sender might choose how many nodes to use, whether to use mostly nodes run by her friends, whether to send in the morning or evening, etc. After all, only she knows the value of her anonymity. But this choice also threatens anonymity — different usage patterns can help distinguish and track users.

Limiting choice of system-wide security parameters can protect users by keeping the noise fairly uniform, but introduces inefficiencies; users that don't need as much protection may feel they're wasting resources. Yet we risk anonymity if we let users optimize their behavior. We can't even let users pay for better service or preferential treatment — the hordes in the coach seats are more anonymous than the few in first class.

[12] Many individuals tend to be myopic in their attitude to privacy. They claim they want it but they are not willing to pay for it. While this might reflect a rational assessment of the trade-offs (that is, quite simply, the agents do not value their anonymity highly enough to justify the cost to protect it), it might also reflect "myopic" behavior such as the hyperbolic discounting of future costs associated to the loss of anonymity. See also [1].

This need to pigeonhole users into a few behavior classes conflicts with the fact that real-world users have a continuum of interests and approaches. Reducing options can lead to reduced usability, scaring away the users and leaving a useless anonymity system.

7 Future Work

There are a number of directions for future research:

- Dummy traffic. Dummy traffic increases costs but it also increases anonymity. In this extension we should study bilateral or multilateral contracts between agents, contractually forcing each agent to send to another agent(s) a certain number of messages in each period. With these contracts, if the sending agent does not have enough real messages going through its node, it will have to generate them as dummy traffic in order not to pay a penalty.
- Reliability. As noted above, we should add reliability issues to the model.
- Strategic dishonest nodes. As we discussed, it is probably more economically sound for an agent to be a lazy node than an anonymity-attacking node. Assuming that strategic bad nodes can exist, we should study the incentives to act honestly or dishonestly and the effect on reliability and anonymity.
- Unknown agent types. We should extend the above scenarios further to consider a probability distribution for an agent's guess about another agent's privacy sensitivity.
- Comparison between systems. We should compare mix-net systems to other systems, as well as use the above framework to compare the adoption of systems with different characteristics.
- Exit nodes. We should extend the above analysis to consider specific costs such as the potential costs associated with acting as an exit node.
- Reputation. Reputation can have a powerful impact on the framework above in that it changes the assumption that traffic will distribute uniformly across nodes. We should extend our analysis to study this more formally.
- Information theoretic metric. We should extend the analysis of information theoretic metrics in order to formalize the functional forms in the agent payoff function.

8 Conclusions

We have described the foundations for an economic approach to the study of strong anonymity infrastructures. We focused on the incentives for participants to act as senders and nodes. Our model does not solve the problem of building a more successful system — but it does provide some guidelines for how to think about solving that problem. Much research remains for a more realistic model, but we can already draw some conclusions:

- Systems must attract cover traffic (many low-sensitivity users) before they can attract the high-sensitivity users. Weak security parameters (e.g. smaller batches) may produce *stronger* anonymity by bringing more users. But to attract this cover traffic, they may well have to address the fact that most users do not want (or do not realize they want) anonymity protection.
- High-sensitivity agents have incentive to run nodes, so they can be certain their first hop is honest. There can be an optimal level of free-riding: in some conditions these agents will opt to accept the cost of offering service to others in order to gain cover traffic.
- While there are economic reasons for distributed trust, the deployment of a completely decentralized system might involve coordination costs which make it unfeasible. A central coordination authority to redistribute payments may be more practical, but could provide a trust bottleneck for an adversary to exploit.

Acknowledgments. Work on this paper was supported by ONR. Thanks to John Bashinski, Nick Mathewson, Adam Shostack, Hal Varian, and the anonymous referees for helpful comments.

References

1. Alessandro Acquisti and Hal R. Varian. Conditioning prices on purchase history. mimeo, University of California, Berkeley, 2002. `http://www.sims.berkeley.edu/~acquisti/papers/`.
2. The Anonymizer. `http://www.anonymizer.com/`.
3. Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, 2001.
4. Theodore Bergstrom, Lawrence Blume, and Hal R. Varian. On the private provision of public goods. *Journal of Public Economics*, 29:25–49, 1986.
5. UC Berkeley. SETI@home: Search for Extraterrestrial Intelligence at Home. `http://setiathome.ssl.berkeley.edu/`.
6. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
7. Richard Cornes and Todd Sandler. *The Theory of Externalities, Public Goods and Club Goods*. Cambridge University Press, 1986.
8. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, 2002.
9. Whitfield Diffie and Susan Landau. *Privacy On the Line: The Politics of Wiretapping and Encryption*. MIT Press, 1998.
10. Roger Dingledine, Michael J. Freedman, David Hopwood, and David Molnar. A Reputation System to Increase MIX-net Reliability. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 126–141. Springer-Verlag, LNCS 2137, 2001. `http://www.freehaven.net/papers.html`.
11. Roger Dingledine and Paul Syverson. Reliable MIX Cascade Networks through Reputation. In Matt Blaze, editor, *Financial Cryptography (FC '02)*. Springer-Verlag, LNCS 2357, 2002.

12. John Douceur. The Sybil Attack. In *1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.
13. Drew Fudenberg and David K. Levine. Open-loop and closed-loop equilibria in dynamic games with many players. *Journal of Economic Theory*, 44(1):1–18, February 1988.
14. Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
15. Sanford J. Grossman and Joseph E. Stiglitz. On the impossibility of informationally efficient markets. *American Economic Review*, 70(3):393–408, June 1980.
16. Raph Levien. Advogato's trust metric. `http://www.advogato.org/trust-metric.html`.
17. Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing congestible network resources. *IEEE Journal of Selected Areas in Communications*, 13(7):1141–1149, September 1995.
18. David Mazières and M. Frans Kaashoek. The Design, Implementation and Operation of an Email Pseudonym Server. In 5^{th} *ACM Conference on Computer and Communications Security (CCS'98)*. ACM Press, 1998.
19. Thomas R. Palfrey and Howard Rosenthal. Underestimated probabilities that others free ride: An experimental test. mimeo, California Institute of Technology and Carnegie-Mellon University, 1989.
20. J. F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.
21. Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–110, 1982.
22. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, 2002.
23. Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Information Hiding (IH 2002)*. Springer-Verlag, LNCS 2578, 2002.
24. Stuart G. Stubblebine and Paul F. Syverson. Authentic attributes with fine-grained anonymity protection. In Yair Frankel, editor, *Financial Cryptography (FC 2000)*, pages 276–294. Springer-Verlag, LNCS 1962, 2001.

On the Economics of Anonymity

Alessandro Acquisti[1], Roger Dingledine[2], and Paul Syverson[3]

[1] SIMS, UC Berkeley acquisti@sims.berkeley.edu
[2] The Free Haven Project arma@mit.edu
[3] Naval Research Lab syverson@itd.nrl.navy.mil

Abstract. Decentralized anonymity infrastructures are still not in wide use today. While there are technical barriers to a secure robust design, our lack of understanding of the incentives to participate in such systems remains a major roadblock. Here we explore some reasons why anonymity systems are particularly hard to deploy, enumerate the incentives to participate either as senders or also as nodes, and build a general model to describe the effects of these incentives. We then describe and justify some simplifying assumptions to make the model manageable, and compare optimal strategies for participants based on a variety of scenarios.

Keywords: Anonymity, economics, incentives, decentralized, reputation

1 Introduction

Individuals and organizations need anonymity on the Internet. People want to surf the Web, purchase online, and send email without exposing to others their identities, interests, and activities. Corporate and military organizations must communicate with other organizations without revealing the existence of such communications to competitors and enemies. Firewalls, VPNs, and encryption cannot provide this protection; indeed, Diffie and Landau have noted that traffic analysis is the backbone of communications intelligence, not cryptanalysis [9].

With so many potential users, it might seem that there is a ready market for anonymity services — that is, it should be possible to offer such services and develop a paying customer base. However, with one notable exception (the Anonymizer [2]) commercial offerings in this area have not met with sustained success. We could attribute these failures to market immaturity, and to the current economic climate in general. However, this is not the whole story.

In this paper we explore the incentives of participants to offer and use anonymity services. We set a foundation for understanding and clarifying our speculations about the influences and interactions of these incentives. Ultimately we aim to learn how to align incentives to create an economically workable system for users and infrastructure operators.

Section 2 gives an overview of the ideas behind our model. Section 3 goes on to describe the variety of (often conflicting) incentives and to build a general model that incorporates many of them. In Section 4 we give some simplifying

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 84–102, 2003.
© Springer-Verlag Berlin Heidelberg 2003

assumptions and draw conclusions about certain scenarios. Sections 5 and 6 describe some alternate approaches to incentives, and problems we encounter in designing and deploying strong anonymity systems.

2 The Economics of Anonymity

Single-hop web proxies like the Anonymizer protect end users from simple threats like profile-creating websites. On the other hand, users of such commercial proxies are forced to trust them to protect traffic information. Many users, particularly large organizations, are rightly hesitant to use an anonymity infrastructure they do not control. However, on an open network such as the Internet, running one's own system won't work: a system that carries traffic for only one organization will not hide the traffic entering and leaving that organization. Nodes must carry traffic from others to provide cover. The only viable solution is to distribute trust. That is, each party can choose to run a node in a shared infrastructure, if its incentives are large enough to support the associated costs. Users with more modest budgets or shorter-term interest in the system also benefit from this decentralized model, because they can be confident that a few colluding nodes are unlikely to uncover their anonymity.

Today, however, few people or organizations are willing to run these nodes. In addition to the complexities of configuring current anonymity software, running a node costs a significant amount of bandwidth and processing power, most of which is used by 'freeloading' users who do not themselves run nodes. Moreover, when administrators are faced with abuse complaints concerning illegal or antisocial use of their systems, the very anonymity that they're providing precludes the usual solution of suspending users or otherwise holding them accountable.

Unlike confidentiality (encryption), anonymity cannot be created by the sender or receiver. Alice cannot decide by herself to send anonymous messages — she must trust the infrastructure to provide protection, and others must use the same infrastructure. Anonymity systems use messages to hide messages: senders are consumers of anonymity and also providers of the cover traffic that creates anonymity for others. Thus users are better off on crowded systems because of the noise other users provide.

Because high traffic is necessary for strong anonymity, agents must balance their incentives to find a common equilibrium, rather than each using a system of their own. The high traffic they create together also enables better performance: a system that processes only light traffic must delay messages to achieve adequately large anonymity sets. But systems that process the most traffic do not necessarily provide the best hiding: if trust is not well distributed, a high volume system is vulnerable to insiders and attackers who target the trust bottlenecks.

Anonymity systems face a surprisingly wide variety of direct anonymity-breaking attacks [3,20]. Additionally, adversaries can also attack the efficiency or reliability of nodes, or try to increase the cost of running nodes. All of these factors combine to threaten the *anonymity* of the system. As Back et al. point out, "in anonymity systems usability, efficiency, reliability and cost become *secu-*

rity objectives because they affect the size of the user base which in turn affects the degree of anonymity it is possible to achieve." [3]

We must balance all of these tradeoffs while we examine the incentives for users and node operators to participate in the system.

3 Analytic Framework

In this section and those that follow, we formalize the economic analysis of why people might choose to send messages through mix-nets.[1] We discuss the incentives for agents to participate either as senders or also as nodes, and we propose a general framework to analyze these incentives. In the next section we consider various applications of our framework, and then in Section 5 we examine alternate incentive mechanisms.

We begin with two assumptions: the agents want to send messages to other parties, and the agents value their anonymity. How various agents might value their anonymity will be discussed below.

An agent i (where $i = (1, ..., n)$ and n is the number of potential participants in the mix-net) bases her strategy on the following possible actions a_i:

1. Act as a user of the system, specifically by sending (and receiving) her own traffic over the system, a_i^s, and/or agreeing to receive dummy traffic through the system, a_i^r. (*Dummy traffic* is traffic whose only purpose is to obscure actual traffic patterns.)
2. Act as an honest node, a_i^h, by receiving and forwarding traffic (and possibly acting as an exit node), keeping messages secret, and possibly creating dummy traffic.
3. Act as a dishonest node, a_i^d, by pretending to forward traffic but not doing so, by pretending to create dummy traffic but not doing so (or sending dummy traffic easily recognizable as such), or by eavesdropping traffic to compromise the anonymity of the system.
4. Send messages through conventional non-anonymous channels, a_i^n, or send no messages at all.

Various benefits and costs are associated with each agent's action and the simultaneous actions of the other agents. The expected benefits include:

1. Expected benefits from sending messages anonymously. We model them as a function of the subjective value each agent i places on the information successfully arriving at its destination, v_{r_i}; the subjective value of keeping her identity anonymous, v_{a_i}; the perceived level of anonymity in the system, p_{a_i} (the subjective probability that the sender and message will remain anonymous); and the perceived level of reliability in the system, p_{r_i} (the subjective probability that the message will be delivered). The subjective value

[1] Mixes were introduced by David Chaum (see [6]). A mix takes in a batch of messages, changes their appearance, and sends them out in a new order, thus obscuring the relation of incoming to outgoing messages.

of maintaining anonymity could be related to the profits the agent expects to make by keeping that information anonymous, or the losses the agents expects to avoid by keeping that information anonymous. We represent the level of anonymity in the system as a function of the traffic (number of agents sending messages in the system, n_s), the number of nodes (number of agents acting as honest nodes, n_h, and as dishonest nodes, n_d), and the decisions of the agent. We assume the existence of a function that maps these factors into a probability measure $p \in [0, 1]$.[2] In particular:

- The level of anonymity of the system is positively correlated to the number of users of the system.
- Acting as an honest node improves anonymity. Senders who do not run a node may accidentally choose a dishonest node as their first hop, significantly decreasing their anonymity (especially in low-latency anonymity systems where end-to-end timing attacks are very hard to prevent [3]). Further, agents who run a node can undetectably blend their message into their node's traffic, so an observer cannot know when the message is sent.
- The relation between the number of nodes and the probability of remaining anonymous might not be monotonic. For a given amount of traffic, sensitive agents might want fewer nodes in order to maintain large anonymity sets. But if some nodes are dishonest, users may prefer more honest nodes (to increase the chance that messages go through honest nodes). Agents that act as nodes may prefer fewer nodes, to maintain larger anonymity sets at their particular node. Hence the probability of remaining anonymous is inversely related to the number of nodes but positively related to the ratio of honest/dishonest nodes. (On the other hand, improving anonymity by reducing the number of nodes can be taken too far — a system with only one node may be easier to monitor and attack. See Section 5 for more discussion.)

If we assume that honest nodes always deliver messages that go through them, the level of reliability in the system is then an inverse function of the share of dishonest nodes in the system, n_d/n_h.

2. Benefits of acting as a node (nodes might be rewarded for forwarding traffic or for creating dummy traffic), b_h.
3. Benefits of acting as a dishonest node (from disrupting service or by using the information that passes through them), b_d.

The possible expected costs include:

1. Costs of sending messages through the anonymous system, c_s, or through a non-anonymous system, c_n. These costs can include both direct financial

[2] Information theoretic anonymity metrics [8,22] probably provide better measures of anonymity: such work shows how the level of anonymity achieved by an agent in a mix-net system is associated to the particular structure of the system. But probabilities are more tractable in our analysis, as well as better than the common "anonymity set" representation.

costs such as usage fees, as well as implicit costs such as the time to build and deliver messages, learning curve to get familiar with the system, and delays incurred when using the system. At first these delays through the anonymous system seem positively correlated to the traffic n_s and negatively correlated to the number of nodes n_h. But counterintuitively, more messages per node might instead *decrease* latency because nodes can process batches more often; see Section 5. In addition, when message delivery is guaranteed, a node might always choose a longer route to reduce risk. We could assign a higher c_s to longer routes to reflect the cost of additional delay. We also include here the cost of receiving dummy traffic, c_r.

2. Costs of acting as an honest node, c_h, by receiving and forwarding traffic, creating dummy traffic, or being an exit node (which involves potential exposure to liability from abuses). These costs can be variable or fixed. The fixed costs, for example, are related to the investments necessary to setup the software. The variable costs are often more significant, and are dominated by the costs of traffic passing through the node.
3. Costs of acting as dishonest node, c_d (again carrying traffic; and being exposed as a dishonest node may carry a monetary penalty).

In addition to the above costs and benefits, there are also *reputation* costs and benefits from: being observed to send or receive anonymous messages, being perceived to act as a reliable node, and being thought to act as a dishonest node.

Some of these reputation costs and benefits could be modelled endogenously (e.g., being perceived as an honest node brings that node more traffic, and therefore more possibilities to hide that node's messages; similarly, being perceived as a dishonest node might bring traffic away from that node). In this case, they would enter the payoff functions only indirectly through other parameters (such as the probability of remaining anonymous) and the changes they provoke in the behavior of the agents. In other cases, reputation costs and benefits might be valued *per se*. While we do not consider either of these options in the simplified model below, Sections 5 and 6 discuss the impact of reputation on the model.

We assume that agents want to maximize their expected payoff, which is a function of expected benefits minus expected costs. Let S_i denote the set of strategies available to agent i, and s_i a certain member of that set. Each strategy s_i is based on the the actions a_i discussed above. The combination of strategies $(s_1, ..., s_n)$, one for each agent who participates in the system, determines the outcome of a game as well as the associated payoff for each agent. Hence, for each complete strategy profile $s = (s_1, ..., s_n)$ each agent receives the expected payoff $u_i(s)$ through the payoff function $u(.)$. We represent the payoff function for each agent i in the following form:

$$u_i = u \begin{pmatrix} \theta\left[\gamma\left(v_{r_i}, p_{r_i}\left(n_h, n_d, a_h^s\right)\right), \partial\left(v_{a_i}, p_{a_i}\left(n_s, n_h, n_d, a_h^s\right)\right), a_i^s\right] + b_h a_i^h + b_d a_i^d \\ -c_s\left(n_s, n_h\right) a_i^s - c_h\left(n_s, n_h, n_d\right) a_i^h - c_d\left(..\right) a_i^d - c_r\left(..\right) a_i^r + (b_n - c_n) a_i^n \end{pmatrix}$$

where $\theta(.), \gamma(.)$, and $\partial(.)$ are unspecified functional forms. The payoff function $u(.)$ includes the costs and benefits for all the possible actions of the agents,

including *not* using the mix-net and instead sending the messages through a non-anonymous channel. We can represent the various strategies by using dummy variables for the various a_i.[3] We note that the probabilities of a message being delivered and a message remaining anonymous are weighted with the values v_{r_i}, v_{a_i}, respectively. This is because different agents might value anonymity and reliability differently, and because in different scenarios anonymity and reliability for the same agent might have different impacts on her payoff.

In Section 4, we will make a number of assumptions that will allow us to simplify this equation and model certain scenarios. We present here for the reader's convenience a table summarizing those variables that will appear in both the complete and simplified equations, as well as one that describes the variables used only in the more complete equation above.

Variables used in both full and simple payoff equations	
	u_i payoff for agent i
	v_{a_i} disutility i attaches to message exposure
	p_a simple case: $p_{a_i} = p_a$ for all i. See next table.
number of nodes (other than i) in mix-net	n_s sending agents (sending nodes)
	n_h honest nodes
	n_d dishonest nodes
dummy variables: 1 if true, 0 otherwise	a_i^h i is an honest node and sending agent
	a_i^s i sends through the mix-net
costs	c_h of running an honest node
	c_s of sending a message through the mix-net

Variables used only in full payoff equation	
	v_{r_i} value i attaches to sent message being received
	p_{a_i} prob. for i that a sent message loses anonymity
	p_r prob. that message sent through mix-net is received
benefits	b_h of running an honest node
	b_d of running a dishonest node
	b_n of sending a message around the mix-net
dummy variables	a_i^d i runs a dishonest node
	a_i^n i sends message around the mix-net
	a_i^r i receives dummy traffic
costs	c_d of running a dishonest node
	c_r of receiving dummy traffic
	c_n of sending a message around the mix-net

Note also that the costs and benefits from sending the message could be distinct from the costs and benefits from keeping the *information* anonymous. For example, when Alice anonymously purchases a book, she gains a profit equal

[3] For example, if the agent chooses not to send the message anonymously, the probability of remaining anonymous p_{a_i} will be equal to zero, $a^{s,d,r,h}$ will be zero too, and the only cost in the function will be c_n.

to the difference between her valuation of the book and its price. But if her anonymity is compromised during the process, she could incur losses (or miss profits) completely independent from the price of the book or her valuation of it. The payoff function $u(.)$ above allows us to represent the duality implicit in all privacy issues, as well as the distinction between the value of sending a message and the value of keeping it anonymous:

Anonymity	*Reliability*
Benefit from remaining anonymous / cost avoided by remaining anonymous, or	Benefit in sending message that will be received / cost avoided by sending such a message, or
Cost from losing anonymity / profits missed because of loss of anonymity	Cost from a message not being received / profits missed by message not being received

Henceforth, we will consider the direct benefits or losses rather than their dual opportunity costs or avoided costs. Nevertheless, the above representation allows us to formalize the various possible combinations. For example, if a certain message is sent to gain some benefit, but anonymity must be protected in order to avoid losses, then v_{r_i} will be positive while v_{a_i} will be negative and p_{a_i} will enter the payoff function as $(1 - p_{a_i})$. On the other side, if the agent must send a certain message to avoid some losses but anonymity ensures her some benefits, then v_{r_i} will be negative and p_{r_i} will enter the payoff function as $(1 - p_{r_i})$, while v_{a_i} will be positive.[4]

With this framework we can compare, for example, the losses due to compromised anonymity to the costs of protecting it. An agent will decide to protect herself by spending a certain amount if the amount spent in defense plus the expected losses for losing anonymity after the investment are less than the expected losses from not sending the message at all.

4 Applying the Model

In this section we apply the above framework to simple scenarios. We make a number of assumptions to let us model the behavior of mix-net participants as players in a repeated-game, simultaneous-move game-theoretic framework. Thus we can analyze the economic justifications for the various choices of the participants, and compare design approaches to mix-net systems.

Consider a set of n_s agents interested in sending anonymous communications. Imagine that there is only one system which can be used to send anonymous messages, and one other system to send non-anonymous messages. Each agent has three options: only send her own messages through the mix-net; send her messages but also act as a node forwarding messages from other users; or don't use the system at all (by sending a message without anonymity, or by not sending

[4] Being certain of staying anonymous would therefore eliminate the risk of v_{a_i}, while being certain of losing anonymity would impose on the agent the full cost v_{a_i}. Similarly, guaranteed delivery will eliminate the risk of losing v_{r_i}, while delivery failure will impose the full cost v_{r_i}.

the message). Thus initially we do not consider the strategy of choosing to be a bad node, or additional honest strategies like creating and receiving dummy traffic.

We represent the game as a simultaneous-move, repeated game because of the large number of participants and because of the impact of earlier actions on future strategies. A large group will have no discernable or agreeable order for the actions of all participants, so actions can be considered simultaneous. The limited commitment produced by earlier actions allows us to consider a repeated-game scenario.[5] These two considerations suggest against using a sequential approach of the Stackelberg type [14, Ch. 3]. For similar reasons we also avoid a "war of attrition/bargaining model" framework (see for example [21]) where the relative impatience of players plays an important role.

4.1 Adversary

Although strategic agents cannot choose to be bad nodes in this simplified scenario, we still assume there is a percentage of bad nodes and that agents respond to this possibility. Specifically we assume a global passive adversary (GPA) that can observe all traffic on all links (between users and nodes, between nodes, and between nodes or users and recipients). Additionally, we also study the case when the adversary includes some percentage of mix nodes. In choosing strategies agents will attach a subjective probability to arbitrary nodes being compromised — all nodes not run by the agent are assigned the same probability of being compromised. This factor influences their assessment of the anonymity of messages they send. A purely passive adversary is unrealistic in most settings, e.g., it assumes that hostile users never selectively send messages at certain times or over certain routes, and nodes and links never selectively trickle or flood messages [23]. Nonetheless, a *global* passive adversary is still quite strong, and thus a typical starting point of anonymity analyses.

4.2 Honest Agents

If a user only sends messages, the cost of using the anonymous service is c_s. This cost might be higher than using the non-anonymous channel, c_n, because of usage fees, usage hassles, or delays. To keep things simple, we assume that all messages pass through the mix-net in fixed-length free routes, so that we can write c_s as a fixed value, the same for all agents. Users send messages at the same time, and only one message at a time. We also assume that routes are chosen randomly by users, so that traffic is uniformly distributed among the nodes.[6]

If a user decides to be a node, her costs increase with the volume of traffic (we focus here on the traffic-based variable costs). We also assume that all agents know the number of agents using the system and which of them are acting as

[5] In Section 3 we have highlighted that, for both nodes and simpler users, variable costs are more significant than fixed costs.

[6] Reputation considerations might alter this point; see Section 5.

nodes. We also assume that all agents perceive the same level of anonymity in the system based on traffic and number of nodes, hence $p_{a_i} = p_a$ for all i. Finally, we imagine that agents use the system because they want to avoid potential losses from not being anonymous. This subjective sensitivity to anonymity is represented by v_{a_i} (we can initially imagine v_{a_i} as a continuous variable with a certain distribution across all agents; see below). In other words, we initially focus on the goal of remaining anonymous given an adversary that can control some nodes and observe all communications. Other than anonymity, we do not consider any potential benefit or cost, e.g., possible greater reliability, from sending around the mix-net. We later comment on the additional reliability issues.

$$u_i = -v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d, a_i^h\right)\right) - c_s a_i^s - c_h\left(n_s, n_h, n_d\right) a_i^h - v_{a_i} a_i^n$$

Thus each agent i tries to *minimize* the costs of sending messages and the risk of being tracked. The first component is the probability that anonymity will be lost given the number of agents sending messages, the number of them acting as honest and dishonest nodes, and the action a of agent i itself. This chance is weighted by v_{a_i}, the disutility agent i derives from its message being exposed. We also include the costs of sending a message through the mix-net, acting as a node when there are n_s agents sending messages over n_h and n_d nodes, and sending messages through a non-anonymous system, respectively. Each period, a rational agent can compare the payoff coming from each of these three one-period strategies.

Action	Payoff
a_s	$-v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d\right)\right) - c_s$
a_h	$-v_{a_i}\left(1 - p_a\left(n_s, n_h, n_d, a_i^h\right)\right) - c_s - c_h\left(n_s, n_h, n_d\right)$
a_n	$-v_{a_i}$

We do not explicitly allow the agent to choose *not* to send a message at all, which would of course minimize the risk of anonymity compromise. Also, we do not explicitly report the value of sending a successful message. Both are simplifications that do not alter the rest of the analysis.[7]

While this model is simple, it allows us to highlight some of the dynamics that might take place in the decision process of agents willing to use a mix-net. We now consider various versions of this model.

[7] We could insert an action a^0 with a certain disutility or cost from not sending any message, and then solve the problem of minimizing the expected losses. Or, we could insert in the payoff function for actions $a^{s,h,n}$ also the payoff from successfully sending a message compared to not sending it (which could be interpreted also as an opportunity cost), and solve the dual problem of maximizing the expected payoff. Either way, the "exit" strategy for each agent will either be sending a message non-anonymously, or not sending it at all, depending on which option maximizes the expected benefits or minimizes the expected losses. Thereafter, we can simply compare the two other actions (being a user, or being also a node) to the optimal exit strategy.

Myopic Agents. Myopic agents do not consider the long-term consequences of their actions. They simply consider the status of the network and, depending on the payoffs of the one-period game, adopt a certain strategy. Suppose that a new agent with a privacy sensitivity v_{a_i} is considering using a mix-net with (*currently*) n_s users and n_h honest nodes.

Then if

$$\begin{gathered}-v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ <-v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s, \text{ and}\\ -v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ <-v_{a_i}\end{gathered}$$

agent i will choose to become a node in the mix-net. If

$$\begin{gathered}-v_{a_i}\left(1-p_a\left(n_s+1,n_h+1,n_d,a_i^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ >-v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s, \text{ and}\\ -v_{a_i}\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s<-v_{a_i}\end{gathered}$$

then agent i will choose to be a user of the mix-net. Otherwise, i will simply not use the mix-net.

Our goal is to highlight the economic rationale implicit in the above inequalities. In the first case agent i is comparing the benefits of the contribution to her own anonymity of acting as a node to the costs. Acting as a node dramatically increases anonymity, but it will also bring more traffic-related costs to the agent. Agents with high privacy sensitivity (high v_{a_i}) will be more likely to accept the trade-off and become nodes because they risk a lot by losing their anonymity, and because acting as nodes significantly increases their probabilities of remaining anonymous. On the other side, agents with a lower sensitivity to anonymity might decide that the costs or hassle of using the system are too high, and would not send the message (or would use non-anonymous channels).

Strategic Agents: Simple Case. Strategic agents take into consideration the fact that their actions will trigger responses from the other agents.

We start by considering only one-on-one interactions. First we present the case where each agent knows the other agent's type, but we then discuss what happens when there is uncertainty about the other agent's type.

Suppose that each of agent i and agent j considers the other agent's reaction function in her decision process. Then we can summarize the payoff matrix in the following way:[8]

[8] We use parameters to succinctly represent the following expected payoffs:

$$\begin{gathered}A_w=-v_w\left(1-p_a\left(n_s+2,n_h+2,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+2,n_h+2,n_d\right)\\ B_w=-v_w\left(1-p_a\left(n_s+2,n_h+1,n_d\right)\right)-c_s\\ C_w=-v_w\\ D_w=-v_w\left(1-p_a\left(n_s+2,n_h+1,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+2,n_h+1,n_d\right)\\ E_w=-v_w\left(1-p_a\left(n_s+1,n_h+1,n_d,a_w^h\right)\right)-c_s-c_h\left(n_s+1,n_h+1,n_d\right)\\ F_w=-v_w\left(1-p_a\left(n_s+2,n_h,n_d\right)\right)-c_s\\ G_w=-v_w\left(1-p_a\left(n_s+1,n_h,n_d\right)\right)-c_s\end{gathered}$$

Agent i / Agent j	a_j^h	a_j^s	a_j^n
a_i^h	A_i, A_j	D_i, B_j	E_i, C_j
a_i^s	B_i, D_j	F_i, F_j	G_i, C_j
a_i^n	C_i, E_j	C_i, G_j	C_i, C_j

As before, each agent has a trade-off between the cost of traffic and the benefit of traffic when being a node, and a trade-off between having more nodes and fewer nodes. In addition to the previous analysis, now the final outcome also depends on how much each agent knows about whether the other agent is honest, and how much she knows about the other agent's sensitivity to privacy.

Of course, for an explicit solution we need a specific functional form for the probability function.[9] Nevertheless, even at this abstract level of description this framework can be mapped into the model analyzed in [19] where two players decide simultaneously whether to contribute to a public good.

In our model, when for example $v_{a_i} \gg v_{a_j}$ and v_{a_i} is large, the disutility to player i from not using the system or not being a node will be so high that she will decide to be a node even if j might free ride on her. Hence if j values her anonymity, but not that much, the strategies a_i^h,a_j^s can be an equilibrium of the repeated game.

In fact, this model might have equilibria with free-riding even when the other agent's type is unknown. Imagine both agents know that the valuations v_{a_i}, v_{a_j} are drawn independently from a continuous, monotonic probability distribution. Again, when one agent cares about her privacy enough, and/or believes that there is a high probability that the opponent would act as a dishonest node, then the agent will be better off protecting her own interests by becoming a node (again see [19]). Of course the more interesting cases are those when these clear-cut scenarios do not arise, which we consider next.

Strategic Agents: Multi-player Case. Each player now considers the strategic decisions of a vast number of other players. Fudenberg and Levine [13] propose a model where each player plays a large set of identical players, each of which is "infinitesimal", i.e. its actions cannot affect the payoff of the first player. We define the payoff of each player as the average of his payoffs against the distribution of strategies played by the continuum of the other players. In other words, for each agent, we will have: $u_i = \sum_{n_s} u_i\,(a_i, a_{-i})$ where the notation represents the comparison between one specific agent i and all the others. Cooperative solutions with a finite horizon are often not sustainable when the actions of other agents are not observable because, by backward induction, each agent will have an incentive to deviate from the cooperative strategy. As compared to the analysis above with only two agents, now a defection of one agent might

[9] We have seen above, however, that privacy metrics like [8,22] do not directly translate into monotonic probability functions of the type traditionally used in game theory. Furthermore, the actual level of anonymity will depend on the mix-net protocol and topology (synchronous networks will provide larger anonymity sets than asynchronous networks for the same traffic divided among the nodes).

affect only infinitesimally the payoff of the other agents, so the agents might tend not to punish the defector. But then, more agents will tend to deviate and the cooperative equilibrium might collapse. "Defection", in fact, could be acting only as a user and refusing to be a node when the agent starts realizing that there is enough anonymity in the system and she no longer needs to be a node. But if too many agents act this way, the system might break down for lack of nodes, after which everybody would have to resort to non-anonymous channels.

We can consider this to be a "public good with free-riding" type of problem [7]. The novel point from a game-theoretic perspective is that the highly sensitive agents actually *want* some level of free-riding, to provide noise. On the other side, they do not want too much free-riding — for example from highly sensitive types pretending to be agents with low sensitivity — if it involves high traffic costs.

So, under which conditions will a system with many players not implode?

First, a trigger strategy might be agreed upon among the many agents, so that the deviation of one single player might be met by the reaction of all the others (as described in [13]). Of course the only punishment available here is making the system unavailable, which has a cost for all agents. In addition, coordination costs might be prohibitive. This is not a viable strategy.

Second, we must remember that highly sensitive agents, for a given amount of traffic, prefer to be nodes (because anonymity will increase) and prefer to work in systems with fewer nodes (else traffic gets too dispersed and the anonymity sets get too small). So, if v_{a_i} is particularly high, i.e. if the cost of not having anonymity is very high for the most sensitive agents, then they will decide to act as nodes regardless of what the others do. Also, if there are enough agents with lower v_{a_i}, again a "high" type might have an interest in acting alone if its costs of not having anonymity would be too high compared to the costs of handling the traffic of the less sensitive types.

In fact, when the valuations are continuously distributed, this *might* generate equilibria where the agents with the highest valuations v_{a_i} become nodes, and the others, starting with the "marginal" type (the agent indifferent between the benefits she would get from acting as node and the added costs of doing so) provide traffic.[10] This problem can be mapped to the solutions in [4] or [17]. At that point an equilibrium level of free-riding might be reached. This condition can be also compared to [15], where the paradox of informationally efficient markets is described.[11]

The problems start if we consider now a different situation. Rather than having a continuous distribution of valuations v_{a_i}, we consider two types of agents: the agent with a high valuation, $v_{a_i} = v_H$, and the agent with a low valuation, $v_{a_i} = v_L$. We assume that the v_L agents will simply participate sending traffic if the system is cheap enough for them to use (but see Section 6.3), and we also assume this will not pose any problem to the v_H type, which in fact has an

[10] Writing down specific equilibria, again, will first involve choosing appropriate anonymity metrics, which might be system-dependent.

[11] The equilibrium in [15] relies on the "marginal" agent who is indifferent between getting more information about the market and not getting it.

interest in having more traffic. Thus we can focus on the interaction between a subset of users: the identical high-types.

Here the "marginal" argument discussed above might not work, and coordination might be costly. In order to have a scenario where the system is self-sustaining and free, and the agents are of high and low types, the actions of the agents must be visible and the agents themselves must agree to react together to any deviation of a marginal player. In realistic scenarios, however, this will involve very high transaction/coordination costs, and will require an extreme (and possibly unlikely) level of rationality for the agents. This equilibrium will also tend to collapse when the benefits from being a node are not very high compared to the costs. Paradoxically, it also breaks down when an agent trusts another so much that she prefers to delegate away the task of being a node. The above considerations however also hint at other possible solutions to reduce coordination costs. We now consider some other mechanisms that can make these systems economically viable.

5 Alternate Incentive Mechanisms

As the self-organized system might collapse under some of the conditions examined above, we discuss now what economic incentives we can get from alternative mechanisms.

1. *Usage fee.* If participants pay to use the system, the "public good with free-riding" problem turns into a "clubs" scenario. The pricing mechanism must be related to how much the participants expect to use the system or how sensitive they are. Sensitive agents might support the others by offering them limited services for free, because they need their traffic as noise. The Anonymizer offers basic service at low costs to low-sensitivity agents (there is a cost in the delay, the limitation on destination addresses, and the hassle of using the free service), and offers better service for money. With usage fees, the cost of being a node is externalized. A hybrid solution involves distributed trusted nodes, supported through entry fees paid to a central authority and redistributed to the nodes. This was the approach of the Freedom Network from Zero-Knowledge Systems. The network was shut down because they were unable to sell enough clients to cover their costs.
2. *"Special" agents.* Such agents have a payoff function that considers the social value of having an anonymous system or are otherwise paid or supported to provide such service. If these agents are paid, the mechanism becomes similar to the hybrid solution discussed above, except anonymity-sensitive agents, rather than act as nodes, pass the money to a central authority. The central authority redistributes the funding among trusted entities acting as nodes.
3. *Public rankings and reputation.* A higher reputation not only attracts more cover traffic but is also a reward in itself. Just as the statistics pages for seti@home [5] encourage participation, publicly ranking generosity creates an incentive to participate. Although the incentives of public recognition and

public good don't fit in our model very well, we emphasize them because they explain most actual current node operators. As discussed above, reputation can enter the payoff function indirectly or directly (when agents value their reputation as a good itself).

If we publish a list of nodes ordered by safety (based on number of messages passing through the node), the high-sensitivity agents will gravitate to safe nodes, causing more traffic and improving their safety further (and lowering the safety of other nodes). In our model the system will stabilize with one or a few mix nodes. In reality, though, p_a is influenced not just by n_h but also by jurisdictional diversity — a given high-sensitivity sender is happier with a diverse set of mostly busy nodes than with a set of very busy nodes run in the same zone. Also, after some threshold of users, latency will begin to suffer, and the low sensitivity users will go elsewhere, taking away the nice anonymity sets.

More generally, a low-latency node may attract many low-sensitivity agents, and thus counterintuitively provide *better* anonymity than one that waits to batch many messages for greater security.

6 A Few More Roadblocks

6.1 Authentication in a Volunteer Economy

Our discussions so far indicate that it may in fact be plausible to build a strong anonymity infrastructure from a wide-spread group of independent nodes that each want good anonymity for their own purposes. In fact, the more jurisdictionally diverse this group of nodes, the more robust the overall system.

However, volunteers are problems: users don't know the node operators, and don't know whether they can trust them. We can structure system protocols to create better incentives for honest principals and to catch bad performance by others, e.g. by incorporating receipts and trusted witnesses [10], or using a self-regulating topology based on verifying reliability [11]. But even when this is feasible, identifying individuals is a problem. Classic authentication considers whether it's the right entity, but not whether the authenticated parties are distinct from one another. One person may create and control several distinct online identities. This *pseudospoofing* problem [12] is a nightmare when an anonymity infrastructure is scaled to a large, diffuse, peer-to-peer design; it remains one of the main open problems in the design of any decentralized anonymity service. The Advogato trust metric [16] and similar techniques rely on humans to make initial trust decisions, and then bound trust flow over a certification graph. However, so far none of these trust flow approaches have provided a clear solution to the problem. Another potential solution, a global PKI to ensure unique identities [24], is unlikely to emerge any time soon.

6.2 Dishonest Nodes vs. Lazy Nodes

We have primarily focused on the strategic motivations of honest agents, but the motivations of dishonest agents are at least as important. An anonymity-breaking adversary with an adequate budget would do best to provide very good service, possibly also attempting DoS against other high-quality providers. None of the usual metrics of performance and efficiency can identify dishonest nodes. Further, who calculates those metrics and how? If they depend on a centralized trusted authority, the advantages of diffusion are lost. Another approach to breaking anonymity is to simply attack the reliability or perceived reliability of the system — this attack flushes users to a weaker system just as military strikes against underground cables force the enemy to communicate over less secure channels.

On the other hand, when we consider strategic dishonest nodes we must also analyze their motivations as rational agents. A flat-out dishonest agent participates only to compromise anonymity or reliability. In doing so, however, a dishonest agent will have to consider the costs of reaching and maintaining a position from which those attacks are effective — which will probably involve gaining reputation and acting as a node for an extended period of time, a cost if the goal is to generally break reliability. Such adversaries will be in an arms race with protocol developers to stay undetected despite their attacks [11]. The benefits from successful attacks might be financial, as in the case of discovering and using sensitive information or a competitor's service being disrupted; or they could be purely related to personal satisfaction. The costs of being discovered as a dishonest node include rebuilding a new node's worth of reputation; but being noticed and exposed as the adversary may have very serious negative consequences for the attacker itself. (Imagine the public response if an Internet provider were found running dishonest nodes.) Thus, all things considered, it might be that the laws of economics work against the attacker as well.

A "lazy" node, on the other hand, wants to protect her own anonymity, but keeps her costs lower by not forwarding or accepting all of her incoming traffic. By doing so this node decreases the reliability of the system. While this strategy might be sounder than the one of the flat-out dishonest node, it also exposes again the lazy node to the risk of being recognized as a disruptor of the system. In addition, this tactic, by altering the flow of the traffic through her own node, might actually reduce the anonymity of that agent.

Surveys and analysis on actual attacks on actual systems (e.g., [18]) can help determine which forms of attacks are frequent, how dangerous they are, and whether economic incentives or technical answers are the best countermeasures.

6.3 Bootstrapping the System and Perceived Costs

Our models so far have considered the strategic choices of agents facing an already existing mix-net. We might even imagine that the system does not yet exist but that, before the first period of the repeated-game, all the players can

somehow know each other and coordinate to start with one of the cooperative equilibria discussed above.

But this does not sound like a realistic scenario. Hence we must discuss how a mix-net system with distributed trust can come to be. We face a paradox here: agents with high privacy sensitivity want lots of traffic in order to feel secure using the system. They need many participants with lower privacy sensitivities using the system first. The problem lies in the fact that there is no reason to believe the lower sensitivity types are more likely to be early adopters. In addition, their *perceived* costs of using the system might be higher than the real costs[12] — especially when the system is new and not well known — so in the strategic decision process they will decide against using the mix-net at all. Correct marketing seems critical to gaining critical mass in an anonymity system: in hindsight, perhaps Zero-Knowledge Systems would have gotten farther had it placed initial emphasis on usability rather than security.

Note that here again reliability becomes an issue, since we must consider both the benefits from sending a message *and* keeping it anonymous. If the benefits of sending a message are not that high to begin with, then a low sensitivity agent will have fewer incentives to spend anything on the message's anonymity. We can also extend the analysis from our model that considers the costs and benefits of a single system to the comparison of different systems with different costs/benefit characteristics. We comment more on this in the conclusion.

Difficulties in bootstrapping the system and the myopic behavior [1] of some users might make the additional incentive mechanisms discussed in Section 5 preferable to a market-only solution.

6.4 Customization and Preferential Service Are Risky Too

Leaving security decisions up to the user is traditionally a way to transfer cost or liability from the vendor to the customer; but in strong anonymity systems it may be unavoidable. For example, the sender might choose how many nodes to use, whether to use mostly nodes run by her friends, whether to send in the morning or evening, etc. After all, only she knows the value of her anonymity. But this choice also threatens anonymity — different usage patterns can help distinguish and track users.

Limiting choice of system-wide security parameters can protect users by keeping the noise fairly uniform, but introduces inefficiencies; users that don't need as much protection may feel they're wasting resources. Yet we risk anonymity if we let users optimize their behavior. We can't even let users pay for better service or preferential treatment — the hordes in the coach seats are more anonymous than the few in first class.

[12] Many individuals tend to be myopic in their attitude to privacy. They claim they want it but they are not willing to pay for it. While this might reflect a rational assessment of the trade-offs (that is, quite simply, the agents do not value their anonymity highly enough to justify the cost to protect it), it might also reflect "myopic" behavior such as the hyperbolic discounting of future costs associated to the loss of anonymity. See also [1].

This need to pigeonhole users into a few behavior classes conflicts with the fact that real-world users have a continuum of interests and approaches. Reducing options can lead to reduced usability, scaring away the users and leaving a useless anonymity system.

7 Future Work

There are a number of directions for future research:

- Dummy traffic. Dummy traffic increases costs but it also increases anonymity. In this extension we should study bilateral or multilateral contracts between agents, contractually forcing each agent to send to another agent(s) a certain number of messages in each period. With these contracts, if the sending agent does not have enough real messages going through its node, it will have to generate them as dummy traffic in order not to pay a penalty.
- Reliability. As noted above, we should add reliability issues to the model.
- Strategic dishonest nodes. As we discussed, it is probably more economically sound for an agent to be a lazy node than an anonymity-attacking node. Assuming that strategic bad nodes can exist, we should study the incentives to act honestly or dishonestly and the effect on reliability and anonymity.
- Unknown agent types. We should extend the above scenarios further to consider a probability distribution for an agent's guess about another agent's privacy sensitivity.
- Comparison between systems. We should compare mix-net systems to other systems, as well as use the above framework to compare the adoption of systems with different characteristics.
- Exit nodes. We should extend the above analysis to consider specific costs such as the potential costs associated with acting as an exit node.
- Reputation. Reputation can have a powerful impact on the framework above in that it changes the assumption that traffic will distribute uniformly across nodes. We should extend our analysis to study this more formally.
- Information theoretic metric. We should extend the analysis of information theoretic metrics in order to formalize the functional forms in the agent payoff function.

8 Conclusions

We have described the foundations for an economic approach to the study of strong anonymity infrastructures. We focused on the incentives for participants to act as senders and nodes. Our model does not solve the problem of building a more successful system — but it does provide some guidelines for how to think about solving that problem. Much research remains for a more realistic model, but we can already draw some conclusions:

- Systems must attract cover traffic (many low-sensitivity users) before they can attract the high-sensitivity users. Weak security parameters (e.g. smaller batches) may produce *stronger* anonymity by bringing more users. But to attract this cover traffic, they may well have to address the fact that most users do not want (or do not realize they want) anonymity protection.
- High-sensitivity agents have incentive to run nodes, so they can be certain their first hop is honest. There can be an optimal level of free-riding: in some conditions these agents will opt to accept the cost of offering service to others in order to gain cover traffic.
- While there are economic reasons for distributed trust, the deployment of a completely decentralized system might involve coordination costs which make it unfeasible. A central coordination authority to redistribute payments may be more practical, but could provide a trust bottleneck for an adversary to exploit.

Acknowledgments. Work on this paper was supported by ONR. Thanks to John Bashinski, Nick Mathewson, Adam Shostack, Hal Varian, and the anonymous referees for helpful comments.

References

1. Alessandro Acquisti and Hal R. Varian. Conditioning prices on purchase history. mimeo, University of California, Berkeley, 2002. `http://www.sims.berkeley.edu/~acquisti/papers/`.
2. The Anonymizer. `http://www.anonymizer.com/`.
3. Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, 2001.
4. Theodore Bergstrom, Lawrence Blume, and Hal R. Varian. On the private provision of public goods. *Journal of Public Economics*, 29:25–49, 1986.
5. UC Berkeley. SETI@home: Search for Extraterrestrial Intelligence at Home. `http://setiathome.ssl.berkeley.edu/`.
6. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
7. Richard Cornes and Todd Sandler. *The Theory of Externalities, Public Goods and Club Goods*. Cambridge University Press, 1986.
8. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, 2002.
9. Whitfield Diffie and Susan Landau. *Privacy On the Line: The Politics of Wiretapping and Encryption*. MIT Press, 1998.
10. Roger Dingledine, Michael J. Freedman, David Hopwood, and David Molnar. A Reputation System to Increase MIX-net Reliability. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 126–141. Springer-Verlag, LNCS 2137, 2001. `http://www.freehaven.net/papers.html`.
11. Roger Dingledine and Paul Syverson. Reliable MIX Cascade Networks through Reputation. In Matt Blaze, editor, *Financial Cryptography (FC '02)*. Springer-Verlag, LNCS 2357, 2002.

12. John Douceur. The Sybil Attack. In *1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.
13. Drew Fudenberg and David K. Levine. Open-loop and closed-loop equilibria in dynamic games with many players. *Journal of Economic Theory*, 44(1):1–18, February 1988.
14. Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
15. Sanford J. Grossman and Joseph E. Stiglitz. On the impossibility of informationally efficient markets. *American Economic Review*, 70(3):393–408, June 1980.
16. Raph Levien. Advogato's trust metric. `http://www.advogato.org/trust-metric.html`.
17. Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing congestible network resources. *IEEE Journal of Selected Areas in Communications*, 13(7):1141–1149, September 1995.
18. David Mazières and M. Frans Kaashoek. The Design, Implementation and Operation of an Email Pseudonym Server. In 5^{th} *ACM Conference on Computer and Communications Security (CCS'98)*. ACM Press, 1998.
19. Thomas R. Palfrey and Howard Rosenthal. Underestimated probabilities that others free ride: An experimental test. mimeo, California Institute of Technology and Carnegie-Mellon University, 1989.
20. J. F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.
21. Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–110, 1982.
22. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, 2002.
23. Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Information Hiding (IH 2002)*. Springer-Verlag, LNCS 2578, 2002.
24. Stuart G. Stubblebine and Paul F. Syverson. Authentic attributes with fine-grained anonymity protection. In Yair Frankel, editor, *Financial Cryptography (FC 2000)*, pages 276–294. Springer-Verlag, LNCS 1962, 2001.

Squealing Euros: Privacy Protection in RFID-Enabled Banknotes

Ari Juels[1] and Ravikanth Pappu[2]

[1] RSA Laboratories
Bedford, MA 01730, USA
ajuels@rsasecurity.com
[2] ThingMagic, LLC
ravi@thingmagic.com

Abstract. Thanks to their broad international acceptance and availability in high denominations, there is widespread concern that Euro banknotes may provide an attractive new currency for criminal transactions. With this in mind, the European Central Bank has proposed to embed small, radio-frequency-emitting identification (RFID) tags in Euro banknotes by 2005 as a tracking mechanism for law enforcement agencies. The ECB has not disclosed technical details regarding its plan. In this paper, we explore some of the risks to individual privacy that RFID tags embedded in currency may pose if improperly deployed. Acknowledging the severe resource constraints of these tags, we propose a simple and practical system that provides a high degree of privacy assurance. Our scheme involves only elementary cryptography. Its effectiveness depends on a careful separation of the privileges offered by optical vs. radio-frequency contact with banknotes, and full exploitation of the limited access-control capabilities of RFID tags.

Keywords: Banknotes, cryptography, RFID, privacy

1 Introduction

Issued under the aegis of the European Central Bank (ECB), the Euro now serves as a common currency for the second largest economic zone in the world, having supplanted the physical currency of member nations at the beginning of 2002. Among the many intricate policy decisions preceding the introduction of the Euro was the determination of banknote denominations. The ECB opted to issue banknotes up to the relatively high denominations of 200 and 500 Euro. At first, this may appear to be a straightforward decision addressing the convenience of consumers and financial institutions. It could ultimately prove, however, to have far reaching consequences as the Euro becomes a currency of international standing rivaling the U.S. dollar. Even before the introduction of the Euro, concern arose that the 500 Euro banknote might emerge as a magnet for international crime [22]. (Indeed, some economists even accused the ECB of trying to attract black-market activity to Europe as a financial stimulus.) At present,

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 103–121, 2003.
© Springer-Verlag Berlin Heidelberg 2003

the physical currency of choice for international black market transactions is the United States one-hundred-dollar bill. The 500 Euro note, however enjoys the advantage of superior portability. A simple observation is illustrative: enough one-hundred dollar bills to fill a briefcase will, when denominated in 500 Euro notes, fit in a mere handbag.

As an apparent counterpoise to this threat, the ECB has disclosed plans to incorporate Radio Frequency ID (RFID) tags into Euro banknotes by 2005 [23,1]. An RFID tag is a tiny device capable of transmitting a piece of static information across short distances. While the ECB has not revealed specifics, it may be presumed that in this proposal, law enforcement officials would be able to employ monitoring equipment to learn the serial numbers of Euro banknotes surreptitiously at short distances. Deployed at highly trafficked locations such as airports, such a system would permit tracking of currency flows, providing a powerful tool for law enforcement monitoring illegal activity such as money laundering and narcotics trade. The difficulty of creating RFID tags is also seen as a potential deterrent to banknote forgery [21].

In this paper, we consider the impact of the ECB proposal on the privacy of bearers of banknotes carrying RFID tags. In brief, the problem is that RFID-tag readers are increasingly easy and inexpensive to buy. On the other hand, RFID tags are too limited in their capabilities to enforce sophisticated information disclosure policies as have been proposed for fully digital forms of cash (see, e.g., [6,7,9,16] for examples). For example, one possible candidate for deployment in Euro banknotes is the Hitachi μ-chip, which simply transmits a 128-bit serial number [1,21]. If banknotes transmit these serial numbers promiscuously, that is, to anyone possessing a reader, then it is possible for petty criminals easily to detect the presence of banknotes carried by passersby. This would be especially problematic if only high-denomination Euro notes were to be tagged, as might be the case given the additional expense of RFID tags. Worse still, it would possible for anyone to track banknotes, and thus the whereabouts and financial dealings of ordinary citizens. Such tracking would be feasible only at short distances, namely a few meters, but might be done without any knowledge on the part of the victim. The fact that serial numbers contain no consumer information would not provide a guarantee against privacy violations. We give a couple of hypothetical examples here to illustrate the problem:

Example 1. Bar X wishes to sell information about its patrons to local Merchant Y. The bar requires patrons to have their drivers' licenses scanned before they are admitted (ostensibly to verify that they are of legal drinking age). At this time, their names, addresses, and dates of birth are recorded.[1] At the same time, Bar X scans the serial numbers of the RFID tags of banknotes carried by its patrons, thereby establishing a link between identities and serial numbers.

[1] The 2-D barcodes on drivers' licenses in certain states already carry demographic information [3]. The automated harvesting of consumer information by bars and restaurants is an emerging practice. Presumably the information on the front of cards, including the name of the card holder, can be harvested through optical character recognition.

Merchant Y similarly records banknote serial numbers of customers from RFID tags. Bar X sells to Merchant Y the address and birthdate data it has collected over the past few days (over which period of time banknotes are likely not yet to have changed hands). In cases where Bar X and Merchant Y hold common serial numbers, Merchant Y can send mailings directly to customers – indeed, even to those customers who merely enter or pass by Merchant Y's shops without buying anything. Merchant Y can even tailor mailings according to the ages of targeted customers. Patrons of Bar X and Merchant Y might be entirely unaware of the information harvesting described in this example.

Example 2. A private detective wishes to know whether Bob is conducting large-value cash transactions at Carl's store. She surreptitiously intercepts the serial numbers on banknotes withdrawn by Bob and also records the serial numbers of those brought by Carl out of his store. If there is any overlap between sets of numbers, she concludes that Bob has given money to Carl. The private detective might reach the same conclusion if Bob leaves without banknotes that he carried into Carl's store. The private detective might also try to reduce her risk of detection by reading the banknotes of Bob and Carl at separate times, e.g., en route to or from the bank.

A slightly better proposal as regards consumer privacy, and still within the capability of even simple RFID tags, is for banknotes only to transmit serial numbers on receiving a special, static law enforcement key. The problem with this approach is that the key would almost certainly become public knowledge. Such a key would have to be universal, that is, embedded in every law enforcement monitoring device. Moreover, a monitoring device would operate by transmitting the key to target banknotes. This means that the law-enforcement key would need to be transmitted and might thus be easily intercepted. As banknotes cannot be reprogrammed in such cases, this approach seems unworkable.

Another possible approach is to employ a cryptographic form of privacy protection. RFID tags in banknotes could carry and transmit their serial numbers only in encrypted form. This approach is still flawed, however, in that the static ciphertext on a serial number is itself a unique identifier. In other words, the encrypted serial number may itself be viewed as a kind of meta-serial-number, itself permitting the promiscuous tracing of banknotes.

More sophisticated cryptographic solutions for privacy protection are possible in principle. For example, in lieu of a basic RFID tag, banknotes might carry small, clock-bearing devices that only respond to law enforcement queries bearing a digitally signed warrant valid within a certain period of time. Again, however, given the requirements for extremely low cost and size, the RFID tags embedded in banknotes will have to possess much more severely limited capabilities than this. Indeed, the most advanced current generation of cheap, passive RFID tags, such as the Atmel TK5552 carry only about 992 bits of user-accessible

memory, and carry no internal power source [10].[2] These tags are capable of only rudimentary computation, such as bitstring comparisons on keys.

1.1 Our Approach and Goals

We propose an approach to privacy protection of RFID tags that does involve a certain amount of cryptographic design, but is fairly simple. Our solution, moreover, does not require any capabilities beyond the limited ones of the current generation of RFID tags. We assume that intensive cryptographic operations take place in relatively high-powered devices for handling banknotes, rather than in the banknotes themselves. We use public-key encryption of serial numbers (and associated digital signatures) in RFID tags in our scheme, with a corresponding private key stored appropriately by a law enforcement agency.

The basic idea in our proposal is to employ *re-encryption* to cause ciphertexts to change in appearance while the underlying plaintexts, i.e., the encrypted serial numbers, remain the same. We may view the global structure of our scheme as essentially analogous to that of a *mix network*, as introduced by [8]. One crucial difference, however, is that the entities performing re-encryption in our scheme have knowledge of the serial numbers, i.e., the plaintexts. Thus, we do not require any special homomorphic properties from the public-key encryption scheme. The term re-encryption generally refers in the literature to use of such homomorphic properties to enable transformation of a ciphertext value without knowledge of the plaintext. In this paper, though, we employ the term re-encryption with the unorthodox assumption that the plaintext is known.

Re-encryption of ciphertexts addresses the problem of their serving as meta-serial-numbers, thereby enforcing privacy even if banknotes transmit information promiscuously. The re-encryption operation might be performed by shops and retail banks, and even by consumers. Some shops now make use of optical scanning devices for electronic cheque conversion [11]. Devices of similar size and cost can perform exactly the operations required by our scheme for banknote privacy. This approach, though, introduces a couple of problems. First, how do we ensure that re-encryption is performed only at appropriate times and not, e.g., by a malicious passerby? Second, how to we ensure that re-encryption is performed properly, and that banknote holders or handlers are not deceptively embedding false information in RFID tags, or indeed, swapping information between banknotes? We address these two critical problems in this paper.

One of the considerable advantages of our scheme is the flexibility it permits in law enforcement policy. For banknotes with which they have only made contact via RFID, law enforcement can only learn the serial number on performing an asymmetric decryption operation. Because the private decryption key for this operation can be distributed in a threshold manner using standard secret-sharing

[2] Although the specified module size is 5mm × 8mm, this includes an indestructible casing for use in automobiles. The IC itself is about 1mm × 1mm in size, and thus potentially suitable for embedding in banknotes. We cite the Atmel TK5552, though, merely as an example of the existing range of capabilities in RFID tags.

techniques [19], a broad range of policies can be used to restrict access to tracing information. In terms of management of this private key, our scheme may be viewed as a type of key escrow on banknote serial numbers. Serial numbers are quasi-public values, however, unlike the keys that are handled by traditional escrow schemes. Thus we employ rather different tools for creating and verifying the escrowed values to begin with.

Any approach of the kind we describe here – and indeed, we believe, any approach that provides effective privacy for RFID tagged banknotes – must permit fairly widespread alteration of RFID tag information. With this in mind, we now provide a rough enumeration of the properties that we feel a banknote-tracing system based on RFID tagging should provide:

1. **Consumer privacy:** Only law enforcement agencies (and not even the Central Bank) should be able to trace banknotes effectively using information transmitted by RFID tags. This should certainly be the case even if law-enforcement RFID signals are intercepted, and should even hold if law-enforcement field monitoring equipment is captured and successfully reverse engineered. Tracing should only be possible using an appropriately protected private key. We formalize this requirement in section 5.
2. **Strong tracing:** Given interception of valid RFID information from a given banknote, law enforcement should be able to determine the associated serial number.
3. **Minimal infrastructure:** Consumers should require no special equipment for the handling of banknotes. Merchants and banks should require only relatively inexpensive devices for this purpose, and should not require persistent network access. The system should be backward compatable, in the sense that banknotes can, if desired, be used and exchanged without reference to RFID tags.
4. **Forgery resistance:** A forger must at a minimum make optical contact with a banknote in order to be able to forge a copy bearing the same serial number and other data, e.g., associated digital signatures. A forger should be unable to forge new banknotes with previously unseen serial numbers, and should be unable to alter the denomination associated with a given banknote.
5. **Privilege separation:** So as to prevent wayward or malicious tampering with banknote information, RFID tag data should only be alterable given optical contact with banknotes, even if readable though RFID contact alone.
6. **Fraud detection:** If invalid law-enforcement information is written to an RFID tag on a banknote, this should be widely detectable, particularly by any merchant handling the banknote.

1.2 Organization

In section 2, we provide an introduction to RFID tags, describing their characteristics and capabilities. We describe our trust assumptions in section 3 as well as conceptual and cryptographic building blocks. We provide details of

our proposed system in section 4. In section 5, we analyze the security of our scheme, proposing a definition of privacy and also touching on the range of non-cryptographic attacks possible in RFID-enabled systems. We conclude in section 6 with a discussion of some of the practical considerations in deploying our system and some open issues. Formal definitions and additional notes are included in the full version of this paper, available at http://www.ari-juels.com or http://web.media.mit.edu/~pappu/htm/publications.htm.

2 A Primer on RFID Tags

As explained above, an RFID tag is a device capable of transmitting radio-frequency signals, typically for the simple purpose of emitting a static identifier, that is, a uniquely identifying bit-string. In its simplest form, an RFID tag consists of a small silicon integrated circuit adjoined to an antenna, which may be printed on substrate roughly as thin as a piece of paper. Such tags presently cost in the vicinity of $0.50 per unit (U.S.). Thanks to emerging manufacturing techniques, however, the per-unit cost of RFID tags promises to drop to $0.05 or less [17,18] in the next several years. Their physical form, while already quite compact, is in the process of reduction to a slender, paper-thin strip just several centimeters in length. Naturally, the computational capabilities of such small, inexpensive devices is quite constrained. As we will see below, most of the work in any communication with the tag is performed by the tag reader.

The cheap and compact RFID tags suitable for inclusion in banknotes are of a type known as *passive*. Passive tags do not have any internal power sources; rather, they are dependent on the RF field from the tag reader for their power. In a typical scenario, the reader first transmits RF radiation at a given frequency. This powers the tag which, after receiving a sufficient amount of power, modulates the incoming radiation with its stored data. The reader then demodulates and decodes the tag's response to recover the data. Examples of passive RFID tags include electronic article surveillance (or anti-theft) tags embedded in compact discs and books.

There are two other, more heavyweight categories of RFID tags known as *semi-passive* and *active* tags. Semi-passive tags have a battery on board. This allows them to be read from a longer range. Active tags, as distinct from passive and semi-passive tags, are capable of initiating the transmission from their location; they do not require a reader to interrogate them first. The only limitations on the range at which semi-passive and active tags can be read are power and reader sensitivity. A mobile telephone is an example of an active tag; it has a unique identity i.e., the phone number, and is capable of initiating transmission to a base station a long distance away. For size and cost reasons, however, it is impractical to include semi-passive or active tags in banknotes.

There are several bands of the spectrum in which RFID tags operate. These bands are usually regulated by quasi-governmental organizations in various countries. In the U.S., the Federal Communications Commission (FCC) is responsible for regulating all telecommunications by radio, television, wire, satellite and ca-

ble. The most commonly used unlicensed frequencies in RFID are 125 KHz (low frequency or LF), 13.56 MHz (high frequency or HF), 915 MHz (ultra high frequency or UHF), and 2.45 GHz (microwave). To many consumers, LF tags are familiar in the form of small plaques mounted on car windshields for the purpose of automatic toll payment. Although there has been no formal announcement regarding the frequency at which ECB banknote tags will operate, there are several reasons to believe that these tags will operate in the microwave band. Among these reasons are: (1) The ICs of microwave tags are extremely small, allowing them to be manufactured at very low cost; (2) The antennae for these tags are also much smaller than those for tags operating at a lower frequency; and finally (3) These tags can be attached to paper substrates with ease [21].

In general, the necessary size of the tag decreases as the frequency increases, but this leads to a substantial increase in the complexity of the tag reader. Further, the rate at which information is transferred from the tag to the reader is directly proportional to the frequency. As a practical matter, this has implications for the amount of time the tag has to spend in the field of a given reader. The lower the data rate from the tag to the reader, the longer the tag has to spend in the field of the reader. For example, a commercially available tag operating at 125 KHz transmits its ID at 7.8 Kbps. This means that a tag has to remain in the field of the reader for approximately 128 ms. At 13.56 Mhz, the tag-to-reader data rate could be on the order of 50 Kbps, allowing a substantially diminished time in the field to read the same amount of data. At 915 MHz, the tag-to-reader data rate is higher still, on the order of 128 Kbps. The high tag-to-reader data rate assumes greater importance as the amount of information stored on the tag increases.

The small, passive tags suitable for inclusion in banknotes may be constructed with electrically-erasable programmable memory (EEPROM). A typical RFID communication protocol allows the reader to perform several operations on the memory of these tags. The simplest possible commands that the reader can issue are *read* and *write*, wherein the reader simply reads or writes the memory on the tag. Many protocols support anti-collision, that is, the ability for multiple tags with unique identities to be simultaneously read by the same reader. A tag may also receive a *sleep* command which renders it unresponsive to further commands from the reader. This state is maintained until the tag receives a *wake* command, accompanied by a tag-specific key. (*Sleep* is thus a keyed function.) Finally, a tag may be completely deactivated with a *kill* command, which renders the memory completely inaccessible forever. We note that typical passive tags available today have memory capacities of no more than a few kilobits and transmit at a maximum rate of about on the order of 100 Kbps.

Our proposed scheme is based on RFID tag functions at a slightly higher level of sophistication, namely *keyed-read* and *keyed-write*. These are access-control functions applied to particular memory cells. An RFID tag will only permit a keyed-read on a read-protected memory cell if it receives a static secret key, and likewise for write-protected memory cells. The current generation of RFID tags do not include keyed write. These functions, though, may be easily enabled

in the manufacturing process, and are envisioned for near future generations of RFID tags. The Atmel TK5552 [10] is an example of a commercially available tag that supports a majority of these functions.

3 Preliminaries

We have stated our system goals in the list of requirements in section 1. Before presenting details, it is also useful for us to give a loose description of the trust model motivating our construction. In particular, we assume the participation of four entity types, characterized as follows:

1. **Central Bank:** The Central Bank, denoted by $\mathcal{B}$, is the organization empowered to create and issue banknotes. $\mathcal{B}$ also furnishes the digital signatures for banknotes, as we shall see. We assume that the principal security-related aim of the Central Bank is to prevent forgery. Thus, for example, the Central Bank has an interest in issuing banknotes with unique serial numbers and in protecting its digital signing key. We do not, however, assume an interest on the part of the Central Bank in protecting consumer privacy or ensuring effective tracing by law enforcement.
2. **Law Enforcement:** This entity, denoted by $\mathcal{L}$, consists of one or more agencies with an interest in tracing banknote flows. Our system aims to provide a high degree of assurance that the values embedded in banknote RFID tags facilitate this tracing. The law enforcement agency, we assume, wishes to ensure that its privileges are not infringed upon, i.e., that the ability of other entities to trace bills is minimized.
3. **Merchant:** Merchants are entities that handle banknotes, accepting them for payment and perhaps agreeing to anonymize them on behalf of consumers as a free service. We assume that most merchants seek to ensure compliance with law enforcement requirements, that is, that they will report irregularities in banknote information. We consider the possibility that merchants may attempt to compromise consumer privacy. We use $\mathcal{M}$ to denote a merchant, treating $\mathcal{M}$ as a generic label. Retail banks may perform the same range of banknote-handling operations as merchants.
4. **Consumer:** The bearer of a banknote, the consumer, denoted generically by $\mathcal{C}$, has an interest in protecting her own privacy. That is, the consumer seeks to restrict tracing of her banknotes in the highest possible degree. Toward this end, we consider that in some cases, consumers may even breach law enforcement regulations by corrupting the tracing information contained on banknote RFID tags.

3.1 Building Blocks and Concepts

Public-key encryption: Our scheme employs as its basis an arbitrary public-key cryptosystem providing chosen-ciphertext security against adaptive adversaries, of which many are known in the literature. We do not define the notion here,

but instead refer the reader to, e.g., [4] for a discussion of cryptosystem security definitions. As explained above, the idea in our system is to generate a ciphertext C on the serial number S for a given banknote under a public key $PK_{\mathcal{L}}$ generated by $\mathcal{L}$. By employing the corresponding private key, $\mathcal{L}$ can extract S from C. In order to achieve the desired security guarantees, the encryption operation must take as input a randomly generated value known as an *encryption factor*. We let $\mathbf{R}$ denote the set of valid encryption factors for a given security parameter.

Note that our privacy and practicality requirements (1-3) as stated in the introduction to this paper can be fully satisfied with a simple system in which a ciphertext C on a unique serial number S for a given banknote is the only information on the RFID tag. On receiving a bank note, a merchant can optically read the serial number from the note (using a scanning device), encrypt it under the public key $PK_{\mathcal{L}}$, and replace the existing ciphertext with this new one. Under the assumption of chosen-ciphertext security, it is infeasible for an adversary to determine whether the new ciphertext indeed corresponds with the old one. (In isolation, this property is known as semantic security [14].) Re-encryption thus provides the desired assurance of consumer privacy. The problem with this approach is that an attacker can create a ciphertext on any serial number she likes and place it on the RFID tag, causing this false serial number to be propagated. Thus our scheme requires some additional components for testing the validity of tracing information.

Digital signature: Rather than encrypting the serial number S for a given banknote, we instead propose encrypting a digital signature Σ on S produced by the Central Bank. This component of our system addresses requirement 4, namely *forgery resistance*. With use of a digital signature of this kind, an attacker cannot forge serial-number information from scratch for placement on any RFID tag. Our system can in principle accommodate any type of digital signature scheme secure against chosen-message attack, as defined in [15]. We discuss particular choices suitable for the limited memory of RFID tags in section 4.1.

Optical contact vs. RFID contact: The two requirements that are not satisfied by even the use of digital signatures are requirements 5 and 6, those of *privilege separation* and *fraud detection*. Privilege separation is important so as to prevent remote attacks on the banknotes of passersby, i.e., erasure or alteration of their RFID tag information through RF contact alone. Fraud detection is also quite important. Without it, a criminal can swap the information of RFID tags on different banknotes, while law enforcement agents will be unable to detect this type of attack without physical or at least optical contact with the banknote. Indeed, if an attacker plants a ciphertext in a banknote corresponding to an invalid signature, a merchant will be unable to detect this fact, since the merchant cannot decrypt it. Our view, however, is that the ability of merchants to detect invalid ciphertexts is critical in preventing criminal tampering with RFID tags, as law enforcement agencies may not often come in physical contact with individual banknotes in circulation.

To address these problems, we exploit in our system design the availability of two different channels, or data types, which we describe as *optical* and *transmission*. Optical information is simply data printed on a banknote, and presumed to be readable by the devices performing re-encryption, namely the banknote-handling machines of merchants. This information may be encoded in human readable form, and perhaps alternatively in machine-readable form as a 2-D bar code. We assume that it includes a serial number S unique to the banknote and also a unique access-rights key D, whose form we specify later; other information such as the denomination, series, and origin of the note might also be included. By transmission information, we mean the contents of the RFID tag as released upon successful query by an RFID reader. Such a distinction between different types of physical contact with security-system components is not often formally identified in security architecture design, but arises from time to time. Examples include digital-rights management systems for CD-ROMs and the "resurrecting duckling" protocol described in [20].

As explained in our primer on RFID tags in section 2, one of their capabilities is control of read/write access privileges by means of static keys. In our proposed system, we thus restrict access privileges to two memory cells in the RFID tag for a banknote. Privileges for these two cells are protected under the key D, which, as stated above, can only be obtained through optical contact with the banknote. The first protected memory cell is that containing the ciphertext C on the serial number and associated digital signature for the banknote. This cell is universally readable via RF, but its write privileges are keyed to D. Our aim here is to satisfy requirement 5, that of privilege separation, thereby limiting adversarial alteration of the ciphertext C.

In the second protected memory cell is stored the encryption factor r particular to the current ciphertext C in the first memory cell. Both read and write privileges for this second cell are keyed under D. Access to this second cell permits verification of the ciphertext C, and does so without knowledge of the law-enforcement decryption key x, as explained below. Hence, by accessing the contents of this second memory cell, and reading S optically from a banknote, it is possible for a merchant to verify that the ciphertext C is correct. Thus, a merchant making optical contact with a banknote can verify the correctness of law-enforcement information. On the other hand, it is important to deny access to the encryption factor r to parties making RF contact alone with a banknote, as they could otherwise extract the serial number S. Hence, by placing the encryption factor in the second memory cell, we satisfy requirement 6, that of fraud detection, while not undermining the privacy requirements of our scheme. After verifying the correctness of the ciphertext C contained in a banknote she has received, $\mathcal{M}$ may then create a new ciphertext C' with a new, random encryption factor r', and use write access privileges obtained from knowledge of D to overwrite C with C' and r with r'.

Tracing attacks by the Central Bank: As discussed later, if the Central Bank $\mathcal{B}$ does not select a unique access key D for each banknote, this can facilitate an attack whereby the bank determines banknote serial numbers through RF

contact alone. In the extreme case, $\mathcal{B}$ can assign the same key D to every banknote. In this case, $\mathcal{B}$ can successfully determine the re-encryption factor for the ciphertext C read from any banknote via RF contact, and then decrypt the associated serial number. We assume that the Central Bank wishes to ensure against forgery, and therefore assigns a unique serial number S to each banknote. We do not, however, entrust $\mathcal{B}$ with the task of guarding against privacy abuses on its own part. Instead, we have $\mathcal{B}$ compute the key D in a manner such that merchants can verify its correct computation, but such that D still carries a sufficient cryptographic guarantee of uniqueness. To do so, we leverage the uniqueness of serial numbers assigned by $\mathcal{B}$, along with a special property on the digital signature scheme used to sign these serial numbers. This special property, known as *signature uniqueness*, means essentially that a signer cannot produce a single signature that is valid for two messages. Discussion of this property and a formal definition are given in the full version of this paper.

4 Our Scheme

Setup: We let $CS = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ denote a public key cryptosystem with component algorithms performing key generation, encryption, and decryption respectively. We also make use of a digital signature system $DS = (\mathsf{SKG}, \mathsf{Sig}, \mathsf{Ver})$ whose constituent algorithms are key generation, signing, and verification respectively. For formal definitions, see the full version of this paper.

For a security parameter k_1 appropriate for long-term security, bank $\mathcal{B}$ generates a digital signing key pair $(PK_{\mathcal{B}}, SK_{\mathcal{B}}) \leftarrow \mathsf{SKG}(1^{k_1})$. Likewise the law enforcement agency $\mathcal{L}$ generates a cryptosystem key pair $(PK_{\mathcal{L}}, SK_{\mathcal{L}}) \leftarrow \mathsf{KG}(1^{k_1})$. The public keys $PK_{\mathcal{B}}$ and $PK_{\mathcal{L}}$ are published for availability to all participating entities. Also published is a collision-intractable hash function $h : \{0,1\}^* \rightarrow \{0,1\}^{2k_2}$ for an appropriate security parameter k_2. In what follows, we let $\|$ denote bitstring concatenation on plaintexts, and let $\in_R$ denote uniform random selection from a set.

Banknote creation: For every banknote i to be printed, $\mathcal{B}$ selects a unique serial number S_i and computes $\Sigma_i \leftarrow \mathsf{Sig}(SK_{\mathcal{B}}, [S_i \parallel den_i])$. Here, den_i is the banknote denomination, incorporated into the digital signature to prevent attacks involving forgery through alteration of a banknote denomination. A denomination specifier might alternatively be included in S_i. Additionally, $\mathcal{B}$ generates an access key $D_i \in \{0,1\}^{k_2}$ for the note. The key D_i is computed as $h(\Sigma_i)$. The *signature-uniqueness* of the digital signature scheme combined with the collision intractability of h together ensures that D_i is unique to each banknote.[3]

[3] It is important that D_i be computed from Σ_i, rather than S_i. Otherwise an attacker able to guess serial numbers successfully would be able to determine D_i values without even making optical contact with target banknotes. For example, it might be that batches of freshly printed banknotes carry consecutive serial numbers. In this case, an attacker making a withdrawal at a bank would be able to guess the serial numbers of other patrons making withdrawals at roughly the same time. If the D_i

$\mathcal{B}$ prints S_i and Σ_i on the banknote in a manner to facilitate automated optical decipherment by merchant machines, e.g., 2-D barcodes. The serial number S_i might also be printed in human-readable form. Additionally, $\mathcal{B}$ computes the ciphertext C_i as an encryption of Σ_i and S_i. In particular, $\mathcal{B}$ inserts a randomly selected encryption factor r_i into memory cell δ_i and the ciphertext $C_i = \mathsf{Enc}(PK_{\mathcal{L}}, [\Sigma_i \parallel S_i], r_i)$ into memory cell γ_i. It is useful to note that $\mathcal{B}$ need not store access keys or other special information for individual banknotes in order for our scheme to work. The bank may simply record serial numbers and denominations according to its normal policy.

Figure 3 provides a schematic layout of the data incorporated into a banknote in our scheme. For visual clarity, we omit subscripts in this figure.

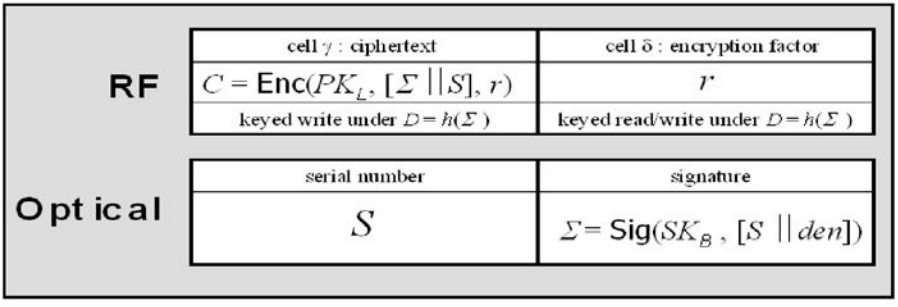

Fig. 1. Banknote data

Banknote verification and anonymization: On receiving a banknote j for payment and/or anonymization, the Merchant $\mathcal{M}$ first verifies the correctness of the existing contents with the following steps:

1. $\mathcal{M}$ optically reads S_j, Σ_j, and D_j.
2. $\mathcal{M}$ computes $D_j = h(\Sigma_j)$.
3. $\mathcal{M}$ reads C_j from γ_j, and performs a keyed read of δ_j under key D_j, yielding the value r_j. If the keyed read fails, the banknote is submitted to law enforcement.
4. $\mathcal{M}$ checks $C_j = \mathsf{Enc}(PK_{\mathcal{L}}, [\Sigma_j \parallel S_j], r_j)$. If not, the invalid ciphertext is reported to law enforcement.

After verifying the contents of the banknote, $\mathcal{M}$ replaces the ciphertext C_j as follows.

5 $\mathcal{M}$ selects $r'_j \in_R \mathbf{R}$ and performs a keyed write of r'_j to δ_j under key D_j. If the keyed write fails, the banknote is submitted to law enforcement.

values of these banknotes are derived from serial numbers, the attacker can track the other patrons via RFID contact.

6 $\mathcal{M}$ computes $C'_j = \mathsf{Enc}(PK_{\mathcal{L}}, [\Sigma_j \parallel S_j], r'_j)$. $\mathcal{M}$ performs a keyed write of C'_j to γ_j under key D_j. $\mathcal{M}$ reports any failure in this step to law enforcement.

Banknote tracing: To obtain the ciphertext C from a target banknote, $\mathcal{L}$ need simply read the contents of the memory cell γ from the associated RFID tag. From the ciphertext C, $\mathcal{L}$ computes the plaintext $[\Sigma \parallel S] = \mathsf{Dec}(SK_{\mathcal{L}}, C)$. Then $\mathcal{L}$ checks whether Σ is a valid signature on S, i.e., whether $\mathsf{Ver}(PK_{\mathcal{B}}, \Sigma, [S \parallel den]) =$ '1'. (The security parameter 1^{k_1} is required here for technical reasons discussed in the full version of the paper.) Provided that C was correct, then this will indeed be the case, and $\mathcal{L}$ will obtain the serial number S.

4.1 Algorithm and Parameter Choices

An especially attractive choice of encryption scheme CS for our system is the El Gamal cryptosystem [13], thanks primarily to its amenability to encoding over elliptic curves. When computed over appropriately parameterized elliptic curves, El Gamal ciphertexts can offer good security at quite compact sizes – on the order of 40 bytes. This is a useful feature given the limited storage available on RFID tags. Let $\mathcal{G}$ denote an appropriate elliptic-curve-based group with prime order q and published generator P. We assume throughout that all cryptographic operations take place over $\mathcal{G}$. For basic El Gamal encryption, we require a group $\mathcal{G}$ over which the Decision Diffie-Hellman problem is presumed to be hard; for the Fujisaki-Okamoto scheme, discussed below, the requirement on $\mathcal{G}$ can be relaxed to the Computational Diffie-Hellman assumption.

Let $SK_{\mathcal{L}} = x \in_R Z_q$ be a private decryption key held by law-enforcement. The value $PK_{\mathcal{L}} = Y = xP$ is the corresponding, published public encryption key. A message $m \in \{0,1\}^w$ for suitably small w is encrypted under $PK_{\mathcal{L}}$ as follows: $\mathsf{Enc}(PK_{\mathcal{L}}, m, r) = (\alpha, \beta) = (m + rY, rP)$, where $r \in_R Z_q$.

By itself, this form of El Gamal is not secure against adaptive chosen-ciphertext attacks. Provided that CS is a one-way encryption scheme, then it is possible to employ a technique due to Fujisaki-Okamato [12] toward this end. Let $h_1, h_2 : \{0,1\}^* \to \{0,1\}^w$ be two cryptographic hash functions. For public key PK, the Fujisaki-Okamoto system converts a basic encryption scheme Enc on plaintext $m \in \{0,1\}^w$ into a hybrid encryption scheme Enc^* as follows:

$$\mathsf{Enc}^*(PK, m, \sigma) = (\mathsf{Enc}(PK, \sigma, h_1(\sigma \parallel m)), h_2(\sigma) \oplus m),$$

where $\sigma \in_R \{0,1\}^w$ is a random encryption factor. The security of this scheme depends on the random oracle assumption on h_1 and h_2.

Although our system can in principle accommodate essentially any type of digital signature, it is important for practical purposes that the signature be short. A particularly attractive scheme is that of Boneh, Shachem, and Lynn [5], which yields signatures of roughly 20 bytes in length. The security is comparable to that of to ECDSA, for which signatures are about twice as long. This scheme makes use of the Weil pairing on a specially chosen elliptic-curve-based group; a signature consists of a single point on the elliptic curve.

Sample parameters: European Central Bank plans presently call for a total availability of 14.5 billion banknotes [2]. Let us suppose that a maximum of one trillion (slightly less than 2^{40}) banknotes are to be printed over the lifetime of our scheme. Thus a serial number S_i might be encoded as a 40-bit value. For the Boneh *et al.* signature scheme, we might use a GDH (Gap Diffie-Hellman) group over E/F_{3^l} yielding a signature size of 154 bits (with discrete-log security equivalent to that of a subgroup of 151 bits) [5]. Thus, a plaintext $[\Sigma_i \parallel S_i]$ in our scheme would be 194 bits in length. We might therefore let $\mathcal{G}$ be an elliptic-curve-based group of 195-bit order. By employing the Fujisaki-Okamoto variant on El-Gamal with $n = 195$, we then achieve a ciphertext length of 585 bits. The encryption factor r for a ciphertext would be 195 bits in length. Thus the total memory requirement for our scheme would be 780 bits – well less than the 992-bit memory capacity of, e.g., the Atmel TK5552 RFID tag. As noted above, in the case that semantic security is deemed sufficient for the underlying cryptosystem CS, the total memory requirement can be reduced to 585 bits. Use of the QUARTZ digital signature scheme or the McEliece variant would further reduce memory requirements. Other optimizations are possible.

5 Security Analysis

The requirements of strong tracing and forgery resistance – indeed, more generally, requirements 2-6 – are straightforward enough from a cryptographic point of view so that we do not formally define them. For example, requirement 4 is fulfilled by the resistance of the underlying signature scheme to forgery. The requirement of consumer privacy (requirement 1) is somewhat trickier to capture. The crux of the problem is that the key D_i for every banknote is known to the Central Bank $\mathcal{B}$. Additionally, a set of these keys may become known to a merchant $\mathcal{M}$, as they are read in the course of handling a banknote. Banknote-handling machines could be rendered tamper-resistant so as to minimize disclosure of these keys. This, however, is not a foolproof approach, particularly as an unscrupulous merchant can create his own banknote-handling machine.[4]

Thus, the strongest definition of consumer privacy regards an adversary with knowledge of a broad range of D_i values. Good privacy guarantees may still be attainable in this case by observing that an attack involving reading of a banknote using key D_i must take place *on the fly*, i.e., during RFID contact with the banknote. In other words, even if the adversary knows the keys D_i for all banknotes, she must guess which key D_i corresponds to a given, target banknote and then transmit that key to the banknote to learn the encryption factor r_i. Passive RFID tags generally transmit at a maximum rate of around 100 Kbps, as explained in section 2. Thus an adversary can expect to be able

[4] To prevent forgery of such machines, it is possible to create tamper-resistant modules that derive a re-encryption factor r'_j from an embedded, merchant-specific key $\kappa_{\mathcal{M}}$. For example, for banknote j, the machine might compute $r'_j = h'(\kappa_{\mathcal{M}}, S_j)$, where h' is a suitable cryptographic hash function. This would enable law enforcement authorities to detect unauthorized banknote-handling machines.

to make only a small number of on-line guesses in most cases – probably just several dozen with a fairly high-power reader operated against a passerby.

We should note additionally that if the Fujisaki-Okamoto construction is employed for encryption, then even knowledge of the encryption factor r_i for a ciphertext C_i does not immediately yield the corresponding serial number S_i. The serial number can be determined from the pair (C_i, r_i), but requires a potentially expensive brute-force attack. This attack may take place off-line, however, and is within the capabilities of a determined attacker. Thus, this partial concealment of S_i does not provide sufficient security *per se*.

Our definition of privacy aims to capture the capabilities of a very strong adversary. We consider the guessing success of an adversary with knowledge of *all* key values D_i. Additionally, as $\mathcal{B}$ itself might potentially constitute the adversary, we assume that this adversary may choose the digital signing keys and serial numbers of banknotes in the system. Another power we assume on the part of the adversary is that of mounting a chosen-ciphertext attack on the underlying cryptosystem.[5] The adversary may read the ciphertext C_i from a banknote, whereupon the aim of the adversary is to guess the corresponding key D_i. The key D_i in our scheme would yield the encryption factor for the associated ciphertext, and thus the serial number. Alternatively, the adversary might try to guess i or S_i directly, but given our strong assumptions about the power of the adversary, this implies the ability to guess D_i. We characterize the success of an adversary $\mathcal{A}$ in terms of the following experiment. We omit the value *den* here for clarity. Let H denote a family of hash functions and $\stackrel{f,k_2}{\leftarrow} H$ denote selection of a hash function from this family under a (possibly randomized) selection algorithm f and security parameter k_2.

Experiment $\mathsf{S\text{-}guess}(\mathcal{A}, \mathcal{G}, CS, DS, H, f); [k_1, k_2]$

1. The key pair $(PK_{\mathcal{L}}, SK_{\mathcal{L}}) \leftarrow \mathsf{KG}(1^{k_1})$ and hash function $h \stackrel{f,k_2}{\leftarrow} H$ are selected.
2. $\mathcal{A}$ receives as input the pair $(PK_{\mathcal{L}}, h)$.
3. $\mathcal{A}$ outputs a public signing key $PK_{\mathcal{B}}$.
4. $\mathcal{A}$ outputs a sequence $\{(S_i, \Sigma_i)\}_{i=1}^{n}$.
5. If $\mathsf{Ver}(PK_{\mathcal{B}}, S_i, \Sigma_i) =$'0' for any i, or $S_i = S_j$ for any $i \neq j$, then the output of the experiment is '0'.
6. For $i \in_R Z_n$ and $r \in_R \mathbf{R}$, $\mathcal{A}$ is given input $C = \mathsf{Enc}(PK_{\mathcal{L}}, [\Sigma_i \parallel S_i], r)$.
7. $\mathcal{A}$ outputs a guess $\tilde{D}$ at $D_i = h(\Sigma_i)$. If $\tilde{D} = D_i$, the output of the experiment is '1'. Otherwise, it is '0'.

Additionally in this experiment, $\mathcal{A}$ has access to encryption and decryption oracles for $PK_{\mathcal{L}}$ at any time during steps 2-5 on any ciphertext, and subsequently on any ciphertext other than C.

[5] As explained above, by removing the assumption that an adversary can mount an adaptive chosen-ciphertext attack, we can reduce the size of the ciphertext in our system from just over 800 to about 600 bits in practice.

Note that an adversary that simply chooses $\tilde{D} \in_R \{D_i\}_{i=1}^n$ can succeed trivially with probability $1/n$. Thus, for any adversary $\mathcal{A}$, let us define the advantage of the adversary for fixed cryptographic primitive choices to S-guess as:

$$\mathbf{Adv}_{\mathsf{S\text{-}guess}}(\mathcal{A}, k_1, k_2) = \mathsf{pr}[\mathsf{S\text{-}guess}(\mathcal{A}, \mathcal{G}, CS, DS, H, f); [k_1, k_2] = \text{`1'}] - 1/n.$$

Proof of the following claim is outlined in the full version of the paper.

Claim 1: Suppose that CS is a public-key cryptosystem with adaptive chosen-ciphertext security and DS a digital-signature scheme with resistance to adaptive chosen-message attack and signature uniqueness. Further, suppose that the hash function family H under f is collision-resistant.[6] Then the quantity $\max_{\mathcal{A}}[\mathbf{Adv}_{\mathsf{S\text{-}guess}}(\mathcal{A}, k_1, k_2)]$ is negligible when taken over all adversaries with running time polynomial in k_1 and k_2.

Our definition and claim are straightforwardly extensible to the scenario in which $\mathcal{A}$ is permitted multiple guesses at D_i, instead of just one. Another type of adversary worth considering is a casual one that does not have knowledge of any keys D_i. It is clear that such an adversary can determine S_i with overall probability only negligible in k_2, even if permitted a polynomial number of RFID tag queries.

5.1 Further Work: Other Attacks

We have characterized the range of possible cryptographic attacks against consumer privacy in our system. Another potential problem for consumer privacy, as mentioned above, is the fact that RFID tags may betray the presence of Euro notes on a bearer. We do not have a comprehensive solution to this problem. One possible approach is for RFID tags to sit normally in a partially "sleep" state, in which they do not "wake" for transmission unless they receive either D_i or a universal law-enforcement key κ. We have already noted the shortcomings of employing a universal law-enforcement key, but this might still be a useful supplementary privacy-protecting measure. An alternative is to embed RFID tags in banknotes of different denominations - or to provide cheap spoofing tags by banks and shops or wallet manufacturers.

Another range of attacks to consider are possible evasions by consumers, that is, violations of the system requirement of strong tracing. Merchants are capable of detecting invalid ciphertexts in banknotes. It is easily possible for the bearer of a banknote, however, to insert a fake ciphertext into the banknote for use while subject to possible law-enforcement monitoring, e.g., before travelling through public places. The bearer can then reintroduce a valid ciphertext prior to spending or depositing the banknote. Indeed, the fake ciphertext used in this attack might be "lifted" from a passerby. One possibility for mitigating this risk

[6] In practice use of a fixed, standard hash function like SHA-1 would be acceptable. If desired, this hash function can additionally be keyed with a random value bound to each individual banknote, effectively a kind of salt.

is to omit Σ_i from the optical information on the banknote, and to construct ciphertexts so that they can be decrypted using r_i. In this case, an attacker who separates the valid signature Σ_i from a banknote and stores it externally runs some risk of losing the signature and thereby invalidating the banknote if she is not careful. Bank policy might require presentation of some proof of identity in order for invalidated banknotes to be exchanged for valid ones.

The possibility of introducing fake ciphertexts into banknotes results from the write capabilities in our proposed system. A very similar attack, however, would be easy to mount even in a system with RFID tags bearing static information. An attacker might with little difficulty create RF devices with the purpose of transmitting fake serial number information – information that may, again, be obtained from passersby. An even more basic attack is possible in any system employing RFID tags: An attacker can simply shield the tags from discovery. Isolation of banknotes in a Faraday cage would constitute a simple and effective attack of this kind. We stress, therefore, that any form of banknote tracing using RFID tags has shortcomings exploitable by a knowledgeable attacker, and that further work is required to address such problems.

6 Conclusion

We have proposed a banknote system design that appeals to the capabilities of the current generation of RFID tags to achieve stronger consumer privacy. It must be stressed that the system does not provide comprehensive privacy protection. Re-encryption of consumer banknotes by merchants, after all, may not occur conveniently with as high a level of frequency as desired by some consumers. Our proposal does, however, go considerably farther than existing ones toward addressing a fundamental privacy issue.

Our observations may also provide some useful insight into how future RFID-tag architectures can offer enhanced functionality at the hardware level in support of both security and privacy. We would like RFID tags in our system to output read-protected information rapidly on presentation of a correct key D_i. On the other hand, an important feature in protecting consumer privacy in our system is the inability of an attacker to mount a rapid on-line attack involving guessing of D_i. In a sense, RFID tags naturally limit the rate of on-line attacks due to their slow processing and transmission capabilities. Ideally, however, this rate limiting might be improved. For example, an RFID tag might be designed to switch to a low data rate mode while transmitting all publicly available information on presentation of an invalid key D_i, thereby delaying subsequent guessing by an attacker. It is our belief that this feature could be incorporated into RFID tags at little cost.

Acknowledgments. The authors extend their thanks to Burt Kaliski and Markus Jakobsson for their helpful comments.

References

1. Security technology: Where's the smart money? *The Economist*, pages 69–70. 9 February 2002.
2. European Central Bank Euro FAQ, 2002. Euro circulation discussed at http://www.euro.ecb.int/en/section1/frequently/printing.html.
3. Registry of Motor Vehicles reforms: Progress report III, 2002. Available at http://www.state.ma.us/rmv/rmvnews/progrpt3.htm.
4. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO '98*, pages 26–45. Springer-Verlag, 1998. LNCS no. 1462.
5. D. Boneh, H. Shacham, and B. Lynn. Short signatures from the Weil pairing. In *ASIACRYPT '01*, pages 514–532, 2001. LNCS no. 2139.
6. S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *CRYPTO '93*, pages 302–318. Springer-Verlag, 1993. LNCS no. 773.
7. E. Brickell, P. Gemmell, and D. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *SODA '95*, pages 157–166, 1995.
8. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
9. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *CRYPTO '88*, pages 319–327. Springer-Verlag, 1988. LNCS no. 403.
10. Atmel Corporation. Atmel TK5552 data sheet, 2001. Available at http://www.atmel.com/atmel/products/prod227.htm.
11. Epson corporation. Epson cheque-imaging scanner: TM-H6000II with TransScan, 2002. Specifications available at http://pos.epson.com/pointofsale/station_printers/tmh6000iiTransScan.
12. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO '99*, pages 537–554. Springer-Verlag, 1999. LNCS no. 1666.
13. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.
14. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comp. Sys. Sci*, 28(1):270–299, 1984.
15. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
16. M. Jakobsson. *Privacy vs. Authenticity.* PhD thesis, University of California at San Diego, 1997.
17. S. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. Available from http://www.autoidcenter.org/.
18. S. Sarma. Radio-frequency identification systems. In B. Kaliski, editor, *CHES '02*. Springer-Verlag, 2002. To appear.
19. A. Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.
20. F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *7th International Workshop on Security Protocols*, pages 172–194. Springer-Verlag, 1999. LNCS no. 1796.

21. K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An ultra small individual recognition security chip. *IEEE Micro*, 21(6):43–49, 2001.
22. C.P. Wallace. The color of money. *Time Europe*, 158(11). 10 September 2001.
23. J. Yoshida. Euro bank notes to embed RFID chips by 2005. *EE Times*. 19 December 2001. Available at http://www.eetimes.com/story/OEG20011219S0016.

How Much Security Is Enough to Stop a Thief?

The Economics of Outsider Theft via Computer Systems and Networks

Stuart E. Schechter and Michael D. Smith

Harvard University
{stuart,smith}@eecs.harvard.edu

Abstract. We address the question of how much security is required to protect a packaged system, installed in a large number of organizations, from thieves who would exploit a single vulnerability to attack multiple installations. While our work is motivated by the need to help organizations make decisions about how to defend themselves, we also show how they can better protect themselves by helping to protect each other.

Keywords: Security Economics, Threat Models, Theft, Exploits

1 Introduction

Before deploying a new laptop computer or installing a new email program, a prudent organization will want to ensure that that system provides adequate security. An organization can determine the security of a computing system by measuring the cost of finding and exploiting a security vulnerability in that system [1]. This measure, known as the *cost to break*, is most effective when you also know how much security your organization requires. To answer the question of how much security is enough, you must first determine what types of adversaries you may need to defend against and what choices are available to each type of adversary.

To this end we introduce *economic threat modeling* as a tool for understanding adversaries motivated by financial gain. Specifically, we model those thieves outside the target organization who would enter via an unreported vulnerability in one of the target's *packaged systems*, those systems that are replicated and installed in many organizations. This model can then be used to estimate what these thieves are willing to pay for system vulnerabilities and how secure the system needs to be to make theft an unprofitable proposition.

Standardization and the wide-spread use of packaged systems means that an organization must look outside itself to determine how thieves might exploit the packaged systems it uses. An organization cannot consider itself safe simply because it would cost a thief more to find and exploit a new vulnerability in one of the organization's systems than that thief could gain by exploiting the vulnerability. Instead, the organization must consider that the thief evaluates his potential financial gains in a global context; he decides whether it is profitable to

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 122–137, 2003.
© Springer-Verlag Berlin Heidelberg 2003

find and exploit a new vulnerability based on all of the organizations deploying that packaged system. Unless an organization can afford to build every system it uses from scratch, it must also measure its security in a global context.

As this paper demonstrates, the global nature of security also works to the advantage of the organization, since the community of all potential victim organizations has similar interests in thwarting attacks. For example, we argue that one way organizations can lower the damage resulting from unreported vulnerabilities is by sharing information about recent attacks. Monitoring companies and federally funded centers such as CERT can help organizations to collect and disseminate such information.

We make a case for the construction of economic threat models in Section 2. We define outside theft in Section 3 and introduce two subcategories. Serial theft, in which a criminal makes repeated uses of the same exploit to rob one victim after another, is analyzed in Section 4. Parallel theft, in which thieves automate the process of exploiting vulnerabilities in order to attack many systems at once, is addressed in Section 5. Applicability and relevance of our work to other threats is discussed in Section 6. In Section 7 we look at the implications of the results to all the members of the defense. Related work follows in Section 8, and we conclude in Section 9.

2 Economic Threat Models

An economist observing a fisherman and wondering why it is that he fishes might pose the following two questions: How difficult it is for the man to catch fish, and how much are consumers willing to pay for fish? Economic threat models are designed to answer these same basic questions. The fundamental importance of these two questions to understanding security becomes clear when one looks at the fisherman and the consumer from a new perspective—that of a fish.

As the security of the fish depends on the number of people who choose to fish and the resources (rods, lines, nets) at their disposal, the security of a system depends on the number of people who stand to profit from attacking it. As in fishing, the choice to attack depends on what one stands to gain given the costs and resources available.

Traditional threat models help us understand who the adversary may be and what motivates them, but do so in a qualitative, not quantitative manner. To understand where these models fall short, it is important to understand where they fit into the security process. Figure 1 shows a representation of this process, traditionally separated into the steps of prevention, detection, and response, expanded to better detail the prevention process. Working backwards, we see that in order to make the right-sized investment in security we must be able to determine the desired level of security. By quantitatively determining the point at which the costs to a potential attacker outweigh the benefits of attack, we can identify this desired security level. Traditional threat models fall short because they do not provide a quantitative measure of much security is enough to deter a given adversary.

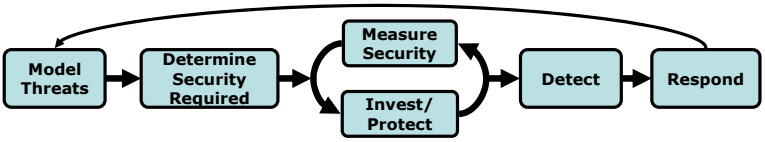

Fig. 1. An illustration of the security process with an emphasis on protective steps.

3 The Threat of Outside Theft

One cannot determine how much security is required to protect against adversaries without making assumptions about who the adversaries are, what motivates them, and what resources are at their disposal. For the next several sections, we focus on thieves outside an organization. *Outside thieves* are individuals or groups of individuals not affiliated with your organization who attack your systems for financial gain. We assume that thieves, in contrast to adversaries such as terrorists, behave rationally in so far that they will not stage attacks that are expected to lead to their own financial loss.

Thieves have many ways to profit from exploiting system vulnerabilities. The most obvious method is for the thief to steal information, such as trade secrets, customer lists, credit card numbers, or cryptographic keys, and then sell that information to others. However, a thief may not actually need to take anything during an attack to create a situation where she can sell something for profit. For example, the thief may create back doors in the systems he attacks and then later sell that access to the highest bidder. Alternatively, the thief may only change the state (e.g. a bank balance) on the target machines. Young and Yung [2] describe a state-changing attack that involves encryption of the data on the victim's machine; the thief profits by ransoming the decryption key.

3.1 Serial Thieves

Serial thieves exploit an unreported vulnerability in a packaged system to attack victim after victim. By concentrating on one (or a small number) of victims at a time, the serial thief can carefully survey the value and location of what each victim is protecting and then maximize the loot obtained while minimizing risk.

A serial thief's crime spree ends when he is caught, when the vulnerability he has learned to exploit has been detected and patched on all target installations, or when the reward of committing another theft with this exploit no longer outweighs the risk of losing the loot already collected.

3.2 Parallel Thieves

Parallel thieves automate their attack to penetrate a large number of targets at the same time. Automation has the clear appeal of multiplicatively increasing the potential loot beyond that obtainable by a more meticulous serial thief in

the same time period. In fact, this may be the only practical way to rob a large number of victims if exploiting the vulnerability is likely to reveal it to the defense and the manufacturer of the vulnerable system is quick to release patches.

There are, however, three costs to automation that relate to our later analysis. First, automation requires the thief to be able to identify the loot without direct human guidance. There is no doubt this can be done (e.g., by looking for files of a particular type), but a parallel thief will not be able to customize the attack to ensure that the maximum value is captured from each victim. Second, automation also forces the thief to make a fixed set of common assumptions about the vulnerability and its surrounding defenses. If the assumptions are incorrect, the attack may fail for a particular organization or worse, may significantly increase the chance that the thief is discovered and later convicted. Finally, it is harder to hide a flood of automated penetrations occurring in a relatively short time period, and many intrusion detection tools look for such statistically significant abnormalities in system usage. This is compounded by the fact that as the number of victims increases, so does the number of transactions required to collect payment for the stolen loot. As we discuss later, there is risk in each of these transactions.

4 Serial Theft

A target's attractiveness to a serial thief is the expected income to the thief from attacking it. This is in turn a function of the probability of success and the amount of loot that may be protected by the target. To simplify the initial analysis, we start by assuming that a thief will attack the most attractive targets, and that there is an unlimited number of homogeneous targets that are all equally attractive. We later extend the analysis by removing this assumption and modelling the unique qualities of each target.

4.1 Homogeneous Targets

We model the choices of the serial thief using a number of variables to represent properties of his environment. The motivation for each crime is the amount of loot that he expects to capture if the theft is successful.

ℓ	Loot, or the value obtained from a successful theft.

For every theft, there is a chance that the thief will be caught, convicted, and punished. We thus define the probability of being convicted and attempt to quantify in dollar terms the value of the punishment.

$\mathcal{P}_c$	Probability of being caught, convicted, and punished for the theft.

F	Fine paid by the thief if convicted, or the dollar equivalent of other punishment levied in the event of conviction, including the value of any confiscated loot.

Catching and convicting a serial thief may be difficult, especially if the thief is in a foreign jurisdiction. Another way to stop a thief who is taking advantage of an unknown vulnerability in a system is to detect how he is breaking in. Each time the thief uses an exploit to break into a system, there is a chance that an intrusion detection system or monitoring firm will be able to observe the thief's means of attack. Once the vulnerability is discovered, intrusion prevention systems may be trained to detect attacks against it and the vulnerable system's manufacturer may distribute a patch. The serial thief will no longer be able to use this exploit to attack organizations that patch their systems.

$\mathcal{P}_d$	Probability that use of the exploit will expose it to the defense and the vulnerability will be patched as a result.

We assume that if the thief is caught and convicted, his methods will be divulged. Thus $\mathcal{P}_c \leq \mathcal{P}_d$.

Finally, there is always a chance that a given attack will fail for reasons other than that the vulnerability was detected and a patch was put into place. Perhaps the target's intrusion detection system had learned just enough from reports of previous attacks at other targets to enable it to recognize a new attack. Perhaps a behavior-based intrusion detection system recognized the theft to be unusual activity and stalled while monitoring personnel were notified.

$\mathcal{P}_f$	Probability that the attack fails, possibly because it is repelled before the criminal can reach the loot.

Note that $\mathcal{P}_f$ and $\mathcal{P}_c$ are independent. An attack may succeed but the thief may be caught and convicted; an attack may fail but the thief may not be captured or convicted; or an attack may succeed and the thief may escape capture or conviction.

To simplify our notation, the probability that the exploit will not be divulged is written $\overline{\mathcal{P}}_d = 1 - \mathcal{P}_d$. The probability that the attack does not fail is written $\overline{\mathcal{P}}_f = 1 - \mathcal{P}_f$.

The expected profit from the ith theft, assuming the exploit has not yet been detected and patched, is the expected loot, $\overline{\mathcal{P}}_f \ell$, minus the expected fine if convicted, $\mathcal{P}_c F$:

$$\overline{\mathcal{P}}_f \ell - \mathcal{P}_c F$$

The expected profit from the ith and all additional thefts is labeled $E_{i\to\infty}$. We account for the additional value of the future thefts by adding the expected value of those thefts on the condition that the exploit can be used again, $\overline{\mathcal{P}}_d E_{i+1\to\infty}$.

$$E_{i\to\infty} = \overline{\mathcal{P}}_f \ell - \mathcal{P}_c F + \overline{\mathcal{P}}_d E_{i+1\to\infty}$$

Expanding reveals a common recurrence of the form $x^0 + x^1 + x^2 + \ldots$, which for $0 < x < 1$ is equal to $\frac{1}{1-x}$.

$$\begin{aligned} E_{i\to\infty} &= \left(\overline{\mathcal{P}}_f \ell - \mathcal{P}_c F\right) \cdot \left(1 + \overline{\mathcal{P}}_d + \overline{\mathcal{P}}_d^2 + \overline{\mathcal{P}}_d^3 + \ldots\right) \\ &= \left(\overline{\mathcal{P}}_f \ell - \mathcal{P}_c F\right) \cdot \frac{1}{1 - \overline{\mathcal{P}}_d} \end{aligned}$$

Thus the value of an exploit to a serial thief attacking homogeneous targets is:

$$E = E_{1\to\infty} = \frac{\overline{\mathcal{P}}_f \ell - \mathcal{P}_c F}{\mathcal{P}_d} \tag{1}$$

This result is quite revealing for the defense. It shows that even if you can't increase the chance of convicting a thief or even thwart attacks that haven't been seen before, you can cut the thief's expected revenue in half by doubling $\mathcal{P}_d$. That is, by doubling the probability that the vulnerability used in the attack will be revealed to the defense and subsequently repaired, the amount of loot that the thief expects to extract from society is reduced by half. Deploying security tools to thwart attacks is also an effective deterrent. Halving the probability that an attack succeeds at each target will also reduce the value of the exploit, by at least if not more than half.

When evaluating different approaches for changing the probabilities in Equation 1, an organization should remember that customized security tools (i.e., ones not readily available to the adversary) are the ones that will be most effective. Unfortunately, these tools are also the most expensive to implement. Security tools that are themselves packaged systems may not be as effective for this purpose. If such packages are available to the thief, he will have had every opportunity to customize the exploit tool in an attempt to circumvent the defense and to avoid detection during an attack. This is a strong argument for using a monitoring company that uses customized detection systems that are unavailable to the adversary.

4.2 Unique Targets

Instead of viewing all targets as homogeneous, we now assume that each is unique. Each potential target organization has two goals in formulating its defense: to minimize the profitability of attack, and to make itself the last target a thief would choose to attack. Action towards the first goal reduces the likelihood that thieves will attack. The second goal is motivated by the fact that if a string of thefts does take place, then the later the organization falls in the set of targets, the more likely it is that the spree will end before the organization is attacked.

We label each target t_i where t_1 is the first target attacked, t_2 the second, and so on. An ordered set of targets $t_1, \ldots, t_i$ is written $\mathbf{T}_i$. The loot for each target is defined as ℓ_i and the punishment for the ith attack is F_i.

$\mathbf{T}_i = t_1, \ldots, t_i$	An ordered set of targets attacked.
ℓ_i	The loot obtained by attacking target t_i.
F_i	The punishment if caught and convicted after the ith attack.

The probability that the vulnerability used to attack the system is detected is defined two different ways.

$\overline{\mathcal{P}}_d(t_i \mid \mathbf{T}_{i-1})$	Probability that, if targets t_1 through t_{i-1} have been attacked, and the exploit has yet to be discovered, that it will not be discovered when t_i is attacked.

$\overline{\mathcal{P}}_d(\mathbf{T}_i)$	Probability that the thief can attack all targets, t_1 through t_i, in order, without the exploit being discovered and fixed.

The latter probability can be expressed inductively in terms of the former: the probability that all i attacks remain undetected is the probability that the first $i-1$ attacks remained undetected multiplied by the probability that the *ith* attack remains undetected.

$$\begin{aligned}
\overline{\mathcal{P}}_d(\mathbf{T}_i) &= \overline{\mathcal{P}}_d(t_i \mid \mathbf{T}_{i-1})\, \overline{\mathcal{P}}_d(\mathbf{T}_{i-1}) \\
\overline{\mathcal{P}}_d(\mathbf{T}_1) &= \overline{\mathcal{P}}_d(t_1 \mid \emptyset) = \overline{\mathcal{P}}_d(t_1) \\
\overline{\mathcal{P}}_d(\mathbf{T}_0) &= \overline{\mathcal{P}}_d(\emptyset) = 1
\end{aligned}$$

We also need functions to describe the probabilities that the thief will or won't be caught and that the attack won't fail.

$\mathcal{P}_c(t_i \mid \mathbf{T}_{i-1})$	Probability that, if targets t_1 through t_{i-1} have already been attacked and the thief was not caught, the thief will be caught and convicted when he attacks t_i.

$\overline{\mathcal{P}}_f(t_i \mid \mathbf{T}_{i-1})$	Probability that, if targets t_1 through t_{i-1} have already been attacked, the attack on t_i will *not* fail to produce loot.

The expected income from attacking unique targets is an extension of the same recurrence shown in the homogeneous target case.

$$\begin{aligned}
E_{1\to n} &= \overline{\mathcal{P}}_f(t_1)\,\ell_1 - \mathcal{P}_c(t_1)\,F_1 + \overline{\mathcal{P}}_d(t_1)\,E_{2\to n} \\
&= \overline{\mathcal{P}}_f(t_1)\,\ell_1 - \mathcal{P}_c(t_1)\,F_1 \\
&\quad + \overline{\mathcal{P}}_d(t_1)\left[\overline{\mathcal{P}}_f(t_2 \mid \mathbf{T}_1)\,\ell_2 - \mathcal{P}_c(t_2 \mid \mathbf{T}_1)\,F_2 + \overline{\mathcal{P}}_d(t_2 \mid \mathbf{T}_1)\,E_{3\to n}\right] \\
&= \overline{\mathcal{P}}_f(t_1)\,\ell_1 - \mathcal{P}_c(t_1)\,F_1 \\
&\quad + \overline{\mathcal{P}}_d(t_1)\left[\overline{\mathcal{P}}_f(t_2 \mid \mathbf{T}_1)\,\ell_2 - \mathcal{P}_c(t_2 \mid \mathbf{T}_1)\,F_2\right] \\
&\quad + \overline{\mathcal{P}}_d(t_1)\,\overline{\mathcal{P}}_d(t_2 \mid \mathbf{T}_1)\,E_{3\to n} \\
&= \overline{\mathcal{P}}_d(\mathbf{T}_0)\left[\overline{\mathcal{P}}_f(t_1 \mid \mathbf{T}_0)\,\ell_1 - \mathcal{P}_c(t_1 \mid \mathbf{T}_0)\,F_1\right] \\
&\quad + \overline{\mathcal{P}}_d(\mathbf{T}_1)\left[\overline{\mathcal{P}}_f(t_2 \mid \mathbf{T}_1)\,\ell_2 - \mathcal{P}_c(t_2 \mid \mathbf{T}_1)\,F_2\right] \\
&\quad + \overline{\mathcal{P}}_d(\mathbf{T}_2)\,E_{3\to n}
\end{aligned}$$

The recurrence simplifies to the following summation where n is the number of thefts.

$$E = E_{1 \to n} = \sum_{i=1}^{n} \overline{\mathcal{P}}_d \left(\mathbf{T}_{i-1}\right) \left[\overline{\mathcal{P}}_f \left(t_i | \mathbf{T}_{i-1}\right) \ell_i - \mathcal{P}_c \left(t_i | \mathbf{T}_{i-1}\right) F_i\right] \tag{2}$$

As long as there are systems for which the thief's chosen vulnerability has yet to be patched, he will attack another target if the next term in Equation 2 is positive, increasing the expected profit. That is, attacking target t_{n+1} increases E so long as:

$$\overline{\mathcal{P}}_f \left(t_{n+1} | \mathbf{T}_n\right) \ell_{n+1} - \mathcal{P}_c \left(t_{n+1} | \mathbf{T}_n\right) F_{n+1} > 0 \tag{3}$$

How can an organization make itself a less attractive target for a serial thief? Most already know that if they have a good relationship with law enforcement, a thief will be more likely to be caught and face conviction. The likelihood of conviction is represented above via $\mathcal{P}_c$.

By taking account of the actions of past victims in our definition of $\mathcal{P}_c$, we can also quantify the effect of providing leads to law enforcement and other members of the defense that may not help the victim directly, but may lead to the thief's capture when others are attacked. Thus we can show how, if a victim organization has a reputation for sharing information to help protect others, it can reduce the expected income to the thief from choosing to include it in the set of targets. As a result, organizations that share information with others make less attractive targets than those that keep information to themselves.

Even if leads do not result in the thief's being caught, they may still reduce the thief's expected income from future attacks. The defense may use information learned from one victim to help protect others or to allow the next victim to learn more about how it has been attacked. Such information might include suspicious traffic patterns or a port number the thief is suspected to have used. We see the results of such efforts in our equations as decreases in $\overline{\mathcal{P}}_d(t|\mathbf{T})$ and $\overline{\mathcal{P}}_f(t|\mathbf{T})$ for potential future targets t.

Similarly, having a reputation for detecting exploits used and cutting off any future revenue from using such an exploit once it is patched will also deter future attacks. It would be foolish for an organization to do all this but neglect to try to foil attacks when they happen. As organizations add intrusion detection and response capabilities, thus increasing $\mathcal{P}_f$, they will also make themselves less attractive targets.

An organization may wish to gauge its attractiveness to a thief in comparison with another organization and pose the question of who would be attacked first. Recall that the later an organization falls in a list of potential victims, the more likely it is that the string of thefts will be brought to an end before the organization is targeted. We can approximate the answer by viewing the world as if there were only two targets, a and b. We write that attacking b followed by a is expected to be more profitable than the reverse ordering by writing:

$$E_{b,a} > E_{a,b}$$

The expected profit from each ordering may be expanded.

$$E_{b,a} = \overline{\mathcal{P}}_f(t_b)\,\ell_b - \mathcal{P}_c(t_b)\,F + \overline{\mathcal{P}}_d(t_b)\left[\overline{\mathcal{P}}_f(t_a|t_b)\,\ell_b - \mathcal{P}_c(t_a|t_b)\,F\right]$$
$$E_{a,b} = \overline{\mathcal{P}}_f(t_a)\,\ell_a - \mathcal{P}_c(t_a)\,F + \overline{\mathcal{P}}_d(t_a)\left[\overline{\mathcal{P}}_f(t_b|t_a)\,\ell_a - \mathcal{P}_c(t_b|t_a)\,F\right]$$

5 Parallel Theft

Thieves undertake parallel attacks in an attempt to penetrate as many systems as possible before the defense can construct a patch for the vulnerability. We assume that the parallel approach ensures that even if the attack is detected, a patch cannot be created and deployed in time to prevent the remaining targets from being penetrated. Hence, the following analysis does not consider $\mathcal{P}_d$, as defined in the previous section.

Thieves benefit from every target attacked in so far as each target increases the potential loot. For n targets, $t_1, t_2, \ldots, t_n$, we define the marginal loot for the ith target as ℓ_i and refer to the total loot as L_n.

ℓ_i	Marginal potential loot from attacking the ith target.

$L_n = \sum_{i=1}^{n} \ell_i$	Total potential loot held by n targets.

As we described earlier, increasing the set of targets also increases the risk of being caught, convicted, and punished. Though detection of a parallel attack may not help the defense at the time of attack, it may prevent the thief from obtaining the loot (for example, if compromised machines are re-secured before the loot is taken) or enable the defense to recover all of the stolen loot at the time of conviction. Thus, a failure or conviction due to attacking any single target will often result in the loss of the loot from all targets. Our analysis incorporates this assumption by setting the total potential loot, L_n, and the total fine paid if caught, F, to be equal.

Given this assumption, we do not want to think about $\mathcal{P}_f$ and $\mathcal{P}_c$ as separate probabilities, but instead simply consider the marginal risk for the ith target as the single quantity r_i. We refer to the probability that the thief successfully captures all of the loot as P_n.

r_i	Marginal increase in the probability that all loot is lost due to the attack on the ith target.

$P_n = 1 - \sum_{i=1}^{n} r_i$	Probability that attack of n targets succeeds and does not lead to detection or conviction.

Using these definitions, we can now state that the expected profit from the attack, E_n, is the potential loot L_n times the probability that the attack succeeds, P_n.

$E_n = P_n \cdot L_n$	Expected profit from the attack.

The expected profit from an additional attack would take into account the marginal loot and marginal risk.

$$E_{n+1} = P_{n+1}L_{n+1} = (P_n - r_{n+1})(L_n + \ell_{n+1})$$

The thief benefits from expanding the set of targets by one if $E_{n+1} > E_n$.

$$\begin{aligned} E_{n+1} &> E_n \\ (P_n - r_{n+1})(L_n + \ell_{n+1}) &> P_n L_n \\ P_n L_n + P_n \ell_{n+1} - r_{n+1} L_n - r_{n+1}\ell_{n+1} &> P_n L_n \\ P_n \ell_{n+1} - r_{n+1}\ell_{n+1} &> r_{n+1} L_n \\ P_{n+1}\ell_{n+1} &> r_{n+1} L_n \end{aligned}$$

The equilibrium point that describes the number of targets attacked, n, is the point at which:

$$P_n \approx \frac{r_n L_{n-1}}{\ell_n} \tag{4}$$

Once the equilibrium point is found, the resulting loot and risk summations, L_n and P_n, can be multiplied to find the expected profit from the attack E_n. Theft is a losing proposition if E_n is less than the cost to find and exploit a new vulnerability. Even if the thief is expected to profit from finding a vulnerability in the system, a given organization may not be targeted if the thief perceives the marginal risk of attacking that organization to be too high.

To help develop intuition for our equilibrium state, consider the case in which all targets contain the same amount of loot ℓ. We can then revise Equation 4 by taking into account that $\ell_n = \ell$ and $L_{n-1} = (n-1)\ell$.

$$\begin{aligned} P_n &\approx \frac{r_n \cdot (n-1)\ell}{\ell} \\ P_n &\approx (n-1) r_n \end{aligned}$$

Dividing by the marginal risk yields the equilibrium state for the case of homogeneous loot:

$$n = \frac{P_n}{r_n} + 1 \tag{5}$$

The conclusion drawn from both the distinct and homogeneous loot cases is the same: maximizing the marginal risk of each additional attack is essential to deterring the parallel thief.

The marginal risk of attack will be low if organizations are defended by homogeneous defenses (packaged systems), as once one system using such a defense is attacked the marginal risk of attacking additional systems that use this defense exclusively will be extremely small. Using a set of systems, each

customized to have unique properties, will be much more effective in keeping marginal risk high. This is not surprising as animal species use a diversity of defense mechanisms, customized in each organism, to survive potent threats from parallel attacks.

Sharing of information is also key to keeping marginal risk high. If the body of knowledge of each member of the defense grows with the number of targets attacked, so will the marginal risk of attack. If organizations do not share information, the body of knowledge of each one will be constant and will not affect marginal risk.

As mentioned above, the thief must not only avoid being traced while breaking in, but must also retrieve the loot and exchange it for payment. To keep his marginal risk low, the thief will require both cash and communications systems that provide the maximum level of anonymity available. Anonymous cash fails to protect the thief if the anonymity can be revoked by law enforcement. Anonymous networks may fail to protect a thief transferring large quantities of stolen data, as transfers of this size are likely to be susceptible to traffic analysis. A thief selling back door access to machines or collecting ransoms takes on additional risk with each transaction. Thieves may try to mitigate these risks by finding ways to group transactions.

Once marginal risk has been addressed, the remaining tool in thwarting the parallel attack is limiting the amount of loot lost to the thief. Backing up data both frequently and securely is an excellent way to limit your exposure to both extortion attacks (cryptoviruses), state changing attacks (such as those that manipulate bank balances and try to cover their tracks), and terrorist attacks. Backup systems, considered by most to be a long solved problem, are ripe for a new era of research against this new class of failure.

6 Other Threats

We focused on outside thieves because their ability to attack multiple targets makes creating an economic model for their behavior a challenge, but a tractable one. To put this work in a larger context we will briefly mention a few additional classes of adversaries and the types of analyses required.

6.1 Insiders

Insiders have unique knowledge and access that yield a considerable advantage in attacking your organization. It is all but unavoidable that there will be partners, users, administrators, and others in or near your organization that require levels of access that make it easier (cheaper) for them to violate your security policies. This is why insider crime is both so dangerous and so common. However, economic models for insider theft are much simpler than for outsiders, as a typical incident will consist of a single crime against a single target. Whereas one cannot simply compare the amount of the loot available with the *cost to break* of a system when protecting from outside theft, such comparisons can be used when defending against an insider.

6.2 Competitors

What makes competitors unique is that they can dramatically benefit from the target organization's losses. As with insiders, this creates an imbalance in cost benefit ratio of attacking one organization (or a small set of organizations) in comparison to others. Like insiders, a competitor's approach can be more easily analyzed by focusing on a small number of players rather than creating models with an unlimited number of targets. When analyzing how competitors might attack, one should not only look at the vulnerability of each system but also at vulnerabilities introduced through system configuration, system interaction, and your security organization. Because few organizations use the same combinations of interacting systems, it may not be possible to amortize the cost of finding vulnerabilities in these configurations. Thus, the cheapest vulnerabilities to find may lie in configuration, system interaction, or organizational weaknesses. These vulnerabilities are most attractive to competitors, as this class of adversary is not as interested in amortizing the cost of finding a vulnerability as they are in attacking a single target.

6.3 Terrorists

As with outside thieves, an analysis of terrorists must assume that they can and will attack a number of victims, amortizing the costs of finding vulnerabilities and exploiting them. Terrorists are particularly dangerous because, like competitors, they perceive benefit from causing damage regardless of whether they are able to retrieve stolen loot. Owing to differences in motivation and jurisdiction, terrorists are likely to believe they have less to lose if detected or caught.

7 Lessons for the Defense

Our analysis above has focused primarily on strategies for organizations that are targets of the thief. These organizations may be aided by many other members of a larger defense.

7.1 Insurers

Target organizations often benefit from transferring their security risks to insurers. While organizations often do not have the expertise to understand these risks, insurers will require that risks be understood before pricing their policies. Insurers also have the power and incentive to force organizations to pay for better security and to provide the knowledge to help them do so.

Insurers benefit when firms share information about attacks, helping to prevent future attacks from succeeding. To foster this sharing of information, insurance companies may want to offer low deductibles and make payment of claims contingent on timely sharing of information. This not only helps prevent future attacks, but as we saw in the analysis in Section 4 this strategy will make the insured systems less attractive targets for attack by serial thieves.

Parallel-theft attacks may expose insurance companies to many concurrent claims, in the same way that an act of God or terrorist attack would. For this reason insurers may want to think twice before insuring against parallel theft. The risk may be greater if the insurer is partnered with a single monitoring firm, as Lloyd's of London is with Counterpane [3], since a single monitoring firm may provide a more homogeneous defense to the insured assets than would a diverse group of firms.

7.2 Monitoring Firms

Monitoring firms are in a unique position not only to detect known attacks, but to discover the first use of new exploits. Clients of monitoring firms that publish information the moment it is discovered will be less attractive targets than clients of monitoring firms that do not.

These firms also are in a unique position to fight parallel attacks through the use of network traffic analysis. They can use this analysis to detect new viruses and worms, to locate the destination of flows of stolen data, and to detect unusual access requests. This is another ripe area for research, as the ability to thwart worms early in the chain of infection would provide immense value to the firm's clients.

Monitoring firms may also benefit from creating or partnering with Honeynets, on whose systems it may be easier to detect parallel attacks. Doing so may improve the monitoring firm's relationship with its insurance partners, or may allow the firm's clients to reduce their insurance premiums.

7.3 Honeynets

Honeypots and Honeynets are closely monitored systems and networks designed by the defense to be infiltrated by the attackers, so that the defense can learn how the attackers is exploiting their systems [4]. Unlike real networks, in which differentiating between an attack and legitimate network use isn't always possible, Honeynets don't have real network traffic. This gives Honeynets a unique advantage in being the first member of the defense to detect a new exploit.

In the name of good citizenship, the creators of Honeynets have used routing rules to keep their compromised systems from being used to attack other machines. This may not be socially optimal. There is no dearth of poorly protected consumer systems available which thieves may turn into 'zombies' used for attack. However, both serial and parallel theft become less profitable with increased risk that a captive machine, used for anonymous routing of stolen goods, may actually be used to detect the exploits (increasing $\mathcal{P}_d$) or link the thief with a crime (increasing $\mathcal{P}_c$). Thus, society may instead want to encourage those running closely monitored Honeynets to allow those systems to be compromised and used by the adversary to stage attacks and route data. In particular, conditions under which Honeynets would receive protection from legal liability should be drafted.

Honeynets are currently run by volunteers. For-profit Honeynets may arise to expand the capabilities of the defense if system vendors offer bounties for reports of newly discovered exploits. Honeynets may also prosper through partnerships with monitoring firms and insurers, or by being integrated into these firms.

7.4 Government

Lawmakers ensure that acts of theft and destruction are matched with deterrent punishments, F. Domestic threats should be countered with federal laws and sentencing guidelines and systems for ensuring cooperation between states. This will end reliance on lax state codes, such as those in Massachusetts, where breaking into a system is a misdemeanor with a maximum punishment of $1,000 and 30 days in prison (Massachusetts General Law, Chapter 266, Section 120F). Once domestic issues are addressed, the true challenge will be to overcome international jurisdictional issues to ensure that $F \neq 0$.

Law enforcement is also essential to fighting theft, especially parallel theft. Whereas serial thieves must contend with the threat that detection will force them to take a loss on their investment in discovering an exploit, the threat of capture after the exploit has done its damage is key to deterring the parallel thief. Once again, jurisdictional issues must be overcome. Lawmakers should discourage the creation of networks and cash systems that have irrevocable anonymity, for if such systems became common the risk of detection in all forms of parallel theft would be greatly reduced. Network traffic monitoring at some level may be necessary to limit the danger of parallel data theft, though such approaches will not be popular with privacy advocates.

8 Related Work

The study of the economics of crime owes a great deal to Becker, who wrote the seminal paper on the subject nearly a quarter century ago [5]. Among his contributions is a model for the expected utility of committing a crime that serves as a foundation for this work. He defines this value by adding to the utility of the crime the product of the utility of the punishment (a negative number) and the probability of conviction.

Ehrlich's work [6] examines both a theory of crime as occupational choice and empirical evidence for this theory. In later work [7] he proposes a market model of criminal offenses, where supply of offenses is a function of the benefits and costs of committing them, including the opportunity cost lost from legitimate labor.

The question of whether it behooves an organization to report information when it is attacked has been posed by Gordon, Loeb, and Lucyshyn [8] in the context of Information Sharing and Analysis Centers (ISACs) such as CERT. A similar question was previously addressed in the context of reporting of household burglaries by Goldberg and Nold [9]. Their empirical analysis found that those households that contained indicators that they would report burglaries

were less likely to be robbed. In Section 4 our model suggests that reporting forensic information discovered in response to a systems attack by a serial thief should help to deter future theft.

Anderson [10] has addressed the unfortunate economics of defending against information terrorists in his seminal paper on the economics of information security. Gordon and Loeb [11] examine the optimal defensive response for an individual acting alone, whereas Varian [12] examines optimal behavior of individuals that make up a collective defense in a variety of collaborative contexts.

Detecting vulnerabilities and releasing patches will only protect those who install the patches. Immediate patching is far from a forgone conclusion. Beattie et al. [13] present a formula for system administrators to determine the optimal time to apply a patch. While the formula itself is straightforward, the inputs required (such as the the probability of attack, cost of recovery, and potential cost of applying a faulty patch) are likely to be the result of speculation. Rescorla [14] presents a case study showing how slowly the user community reacted to patch a serious vulnerability in Apache both before and after a virus exploiting that vulnerability was released. Much still needs to be done to improve the community's patching process before we can assume that the release of a patch will bring a criminal's spree to a halt.

Finally, this paper relies on the assumption that system security can be measured and that this measurement is the cost to acquire a means of breaking into a system. This method was formally developed by us [1,15], but owes much to the work of Camp and Wolfram [16], who first proposed that markets for vulnerabilities be created.

9 Conclusion

We have introduced a model for estimating the value of a system exploit to an outside thief. This model takes into account investments in intrusion detection and response, both internally and by outside monitoring firms. Using the model, an organization can gauge its attractiveness to outside thieves and determine how much security is required in the packaged systems it purchases. Beyond choosing how much to spend on security, analysis of the social aspects of intrusion detection and response strategies, such as information sharing, can be evaluated for their effectiveness in deterring future attacks.

Acknowledgements. This paper could not have been completed without the advice, comments, and suggestions from Fritz Behr, Glenn Holloway, David Malan, David Molnar, and Omri Traub. This research was supported in part by grants from Compaq, HP, IBM, Intel, and Microsoft.

References

1. Schechter, S.E.: Quantitatively differentiating system security. In: The First Workshop on Economics and Information Security. (2002)

2. Young, A., Yung, M.: Cryptovirology: Extortion-based security threats and countermeasures. In: Proceedings of the IEEE Symposium on Security and Privacy. (1996) 129–140
3. Counterpane Internet Security, Lloyd's of London: Counterpane Internet Security announces industry's first broad insurance coverage backed by Lloyd's of London for e-commerce and Internet security. http://www.counterpane.com/pr-lloyds.html (2000)
4. The Honeynet Project: Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community. Addison-Wesley (2001)
5. Becker, G.S.: Crime and punishment: An economic approach. The Journal of Political Economy **76** (1968) 169–217
6. Ehrlich, I.: Participation in illegitimate activities: A theoretical and empirical investigation. The Journal of Political Economy **81** (1973) 521–565
7. Ehrlich, I.: Crime, punishment, and the market for offenses. The Journal of Economic Perspectives **10** (1996) 43–67
8. Gordon, L.A., Loeb, M.P., Lucyshyn, W.: An economics perspective on the sharing of information related to security breaches: Concepts and empirical evidence. In: The First Workshop on Economics and Information Security. (2002)
9. Goldberg, I., Nold, F.C.: Does reporting deter burglars?–an empirical analysis of risk and return in crime. The Review of Economics and Statistics **62** (1980) 424–431
10. Anderson, R.J.: Why information security is hard, an economic perspective. In: 17th Annual Computer Security Applications Conference. (2001)
11. Gordon, L.A., Loeb, M.P.: The economics of information security investment. ACM Transactions on Information and System Security **5** (2002) 438–457
12. Varian, H.R.: System reliability and free riding. In: The First Workshop on Economics and Information Security. (2002)
13. Beattie, S., Arnold, S., Cowan, C., Wagle, P., Wright, C.: Timing the application of security patches for optimal uptime. In: Proceedings of LISA '02: 16th Systems Administration Conference. (2002)
14. Rescorla, E.: Security holes... who cares? http://www.rtfm.com/upgrade.pdf (2002)
15. Schechter, S.E.: How to buy better testing: Using competition to get the most security and robustness for your dollar. In: Proceedings of the Infrastructure Security Conference. (2002)
16. Camp, L.J., Wolfram, C.: Pricing security. In: Proceedings of the CERT Information Survivability Workshop. (2000) 31–39

Cryptanalysis of the OTM Signature Scheme from FC'02

Jacques Stern[1] and Julien P. Stern[2]

[1] Dépt d'Informatique, Ecole normale supérieure, 45 rue d'Ulm,
75230 Paris Cedex 05, France.
Jacques.Stern@ens.fr
[2] Cryptolog International SAS, 16–18 rue Vulpian, 75013 Paris, France.
julien@cryptolog.com

Abstract. At Financial Cryptography 02, Okamoto, Tada, and Miyagi [8] proposed a new fast signature scheme of the Schorr/DSS family, without on line multiplication. Following earlier proposals [5,10,11], a part of the data, independent of the message to sign, is generated at a preprocessing stage, while the computing effort needed to complete the signature "on the fly", is dramatically reduced. Whereas the so-called GPS scheme from [5,10] and its variant from [11] avoid modular operations by computing over the integers, thus reducing the workload to one (regular) multiplication, the new scheme simply gives up multiplication at the cost of bringing back a single modular reduction with respect to a 160 bit integer. Thus, the scheme could appear as achieving better performances. Unfortunately, due to a concealed design weakness, the scheme in [8] is insecure with the proposed parameters. The present paper shows a devastating attack against the scheme, forging a signature in $\simeq 2^{25}$ operations. The scheme can be rescued in a rather straightforward way by significantly raising the parameters, but this degrades its performances which do not compare anymore favorably to [10]. In place, we suggest to replace modular reduction by another novel operation, which we call *dovetailing*. We argue that this operation can be performed in such an efficient way that it could allow for signing with a memory card, rather than a smart card. This equally applies to GPS but the new scheme is better than GPS in terms of signature size.

1 Introduction

In the last twenty years, public key cryptography has dramatically developed, as a means to offer confidentiality to end users. Popular applications such as SSL are now widely accepted as a simple and cost effective method to protect data obtained from WEB servers. However, in many applications, notably outside the Internet, public key techniques are hampered by the computing cost that they require. Currently, banking cards and mobile phones, almost exclusively make use of conventional symmetric cryptography. This entails serious drawbacks such as the impossibility of producing actual digital signatures, or of achieving off-line authentication without storing secrets in the verifying device. Many potential

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 138–148, 2003.
© Springer-Verlag Berlin Heidelberg 2003

applications would greatly benefit of the availability of signature and identification schemes that could be executed, say, on a low cost smart card. The Holy Grail here would be the use of memory cards: this would allow to authenticate phone cards or to sign up at toll booths, thus opening a mass market to public key techniques.

In an attempt to find a partial solution to the above, several authors [3,12, 13] have proposed to break signature generation into two steps

1. a preprocessing step, where data independent of the message to sign are generated,
2. a second step, performed "on the fly", where the signature is completed, with limited computing effort.

Even, Goldreich, and Micali [3] have shown a general albeit not very practical method for converting any signature scheme into a scheme of this form. This has recently been pursued by Shamir and Tauman [14]. In a more restricted setting, the discrete logarithm has been found particularly suitable to this approach, as already noted by Schnorr [12,13]: ignoring one modular addition, the only operation that needs to be performed on the fly is a single modular multiplication. In [5,10], this was reduced to a single regular multiplication. Besides reducing the computing cost, discarding modular multiplication had several nice consequences in terms of key size or signature size. Aiming at a standard security level measured by 2^{80} basic computing steps, the scheme in [10] allowed for secret keys of 160 bits and for signatures of 420 bits.

A further step was taken in a paper [8] presented at Financial Cryptography 2002. In this paper, a scheme that could sign with no on-the-fly multiplication was proposed. We will refer to this scheme as OTM, from the names of the authors, Takeshi Okamoto, Mitsuru Tada and Atsuko Miyaji. Besides avoiding multiplication, the scheme was achieving secret key size 160 bits and signature size 304 bits, thus clearly beating the GPS scheme from [10], at least in terms of size. The price to pay for giving up multiplication was to bring back a single modular reduction with respect to the secret key s, a 160 bit integer. Whether or not the latter is more time consuming than multiplication is debatable and will be discussed further on. In any case, the answer clearly depends on the size of the integer to be reduced modulo s, and any increase of this size entails a loss of efficiency.

In this paper, we first show that, due to a concealed design weakness, the scheme in [8] is insecure with the proposed parameters. The present paper shows a devastating attack against the scheme, forging a signature in $\simeq 2^{25}$ operations. We next discuss whether the scheme can be rescued. A rather straightforward option consists in significantly raising the parameters. However, as explained above, this degrades the performances which do not any more compare favorably to [10]. In place, we suggest to replace modular reduction by another novel operation, which we call *dovetailing*. Basically, dovetailing a pair of integers (r, e) with respect to a smaller integer s is adding to r a small multiple of s so to as to make the trailing bits of the result match with e. Surprisingly, we found that this operation can be performed in an extremely efficient way. We even

argue that the resulting scheme could allow for signing with a memory card, rather than a smart card.

2 A Flaw in the OTM Scheme

2.1 Brief Description of the Scheme

The public key of the OTM scheme consists of an RSA integer n, together with an element g of $\mathbb{Z}_n^\star$, whose order is s is bounded by $S = 2^k$ (typically $S = 2^{160}$). Paper [8] describes how to manufacture such pairs (n, g). We will return to the matter further on in the present paper. The secret key is precisely the order s of g. The signature scheme is derived from an identification scheme, depicted on figure 1. Notations a, ℓ refer to additional parameters of the scheme, typically $a = 224$, $\ell = 24$.

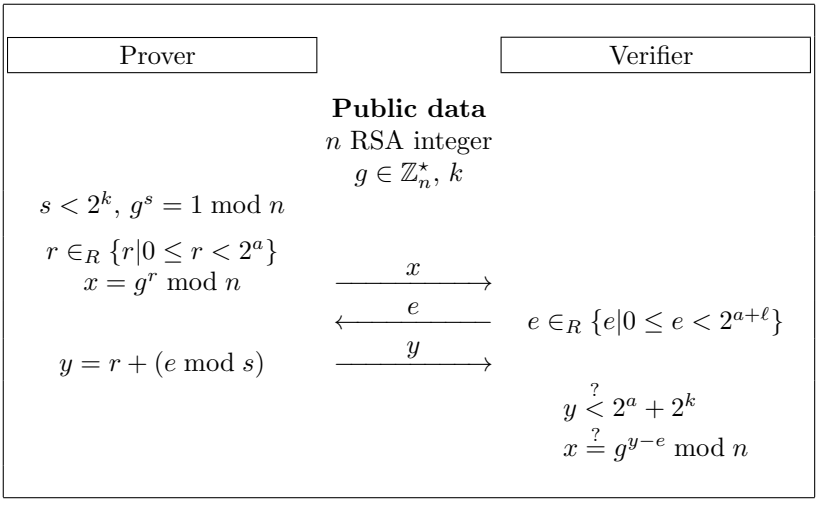

Fig. 1. The OTM identification scheme from FC'02

To derive the signature scheme, one applies the usual trick from [4], replacing e by a hash value $H(m, x)$, computed from the message m and the so-called commitment x. The signature is the pair (e, y). To achieve bit length 304, as claimed in [8], one uses another trick, keeping only say 80 bits of $H(m, x)$ and extending the result to $a + \ell$ bits by means of a hash function again.

2.2 A Forgery Attack

The security analysis of the OTM appearing in [8], relies on the claim that an attacker cannot cheat the identification protocol with probability significantly

larger than $2^{a+\ell}$ (theorem 4 of [8]). This result is incorrect, as shown by the attack that we now mount. The attack is based on guessing the $\ell + 1$ leading bits of the "challenge" e. This means guessing e_1, where

$$e = e_1 2^{a-1} + e_0$$

and $0 \leq e_0 < 2^{a-1}$. From this guess, the attacker manufactures his commitment x by picking at random r, $0 \leq r < 2^{a-1}$ and setting $x = g^{r - e_1 2^{a-1}} \bmod n$. Upon receiving the challenge e, the attacker gives up if the leading bits of e do not match up with e_1. Otherwise, he is successfully identified by returning

$$y = r + e_0$$

Indeed, both tests $y \stackrel{?}{<} 2^a + 2^k$ and $x \stackrel{?}{=} g^{y-e} \bmod n$ are easily seen to be satisfied.

The above directly translates into a forgery attack against the signature scheme. The attacker guesses an $\ell + 1$ bit integer e_1 and computes $H(m_i, x)$ for a sample of messages m_i until the ℓ leading bits of the result match up with e_1. The forgery requires $2^{\ell+1}$ operations.

2.3 Repairing the Scheme

It appears that theorem 4 from [8] is correct when the security bound $1/2^b$ is replaced by $1/2^\ell$. Without going into details, we just mention that the "extractor" that the proof builds needs to find a situation where the attacker can answer two queries with different leading bits, given the same commitment. The proof in [8] is erroneous in that it simply looks for distinct queries.

Thus, the signature scheme can be repaired by raising ℓ to at least 80. At this point, we wish to suggest to raise a to at least $k + 80$. The size of a controls the bound $2s/2^a$ for the statistical zero-knowledge property (theorem 5 of [8]). However, this bound is only for a single execution and should be multiplied by the number of identifications (signatures) allowed. Thus, we feel that the suggested margin of 64 bits is not enough.

We have a final concern about the scheme, related to the key generation algorithm. The RSA integer $n = pq$ used in [8] is manufactured in such a way that $p = 2p'p'' + 1$, and $q = 2q'q'' + 1$, with p', q' prime integers and p'', q'' odd. Next, an element g_p of $\mathbb{Z}_p^\star$ of order p' is built, and similarly an element g_q of $\mathbb{Z}_q^\star$ of order $2q'$. Finally g is obtained by Chinese remaindering. Now, the paper states that taking s of size 160 bits is enough to be immune to Pollard's method [9] for finding the order of g, since the method has computing time $\mathcal{O}(\sqrt{s})$. Thus, g has order $< 2^{80}$ in either $\mathbb{Z}_p^\star$ or $\mathbb{Z}_q^\star$. Although we have not found any attack, we are concerned that revealing such a g might open the way to speeding up the factorization of n. Of course, the OTM scheme can perfectly be adapted to the case where n is a prime number. However, in this case, the order of g is a relatively small factor of $n - 1$, and accordingly it should be taken large enough to withstand ECM factorization. Given the current ECM factoring record of 54 decimal digits (see [7]), one would have to raise the size of s to at least 256 bits.

2.4 Comparing the Resulting Scheme to GPS

The GPS scheme from [5,10] is depicted on figure 2 below.

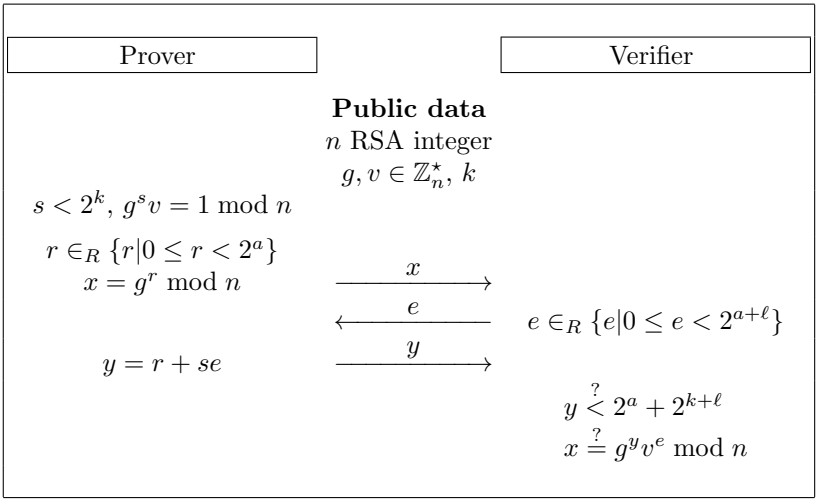

Fig. 2. The GPS identification scheme

Typical parameters in signature mode, where e is replaced by a hash value $H(m, x)$, are $k = 180$, $\ell = 80$, $a = 340$, which yield signature size over 400 bits.

As noticed by the authors of [8], raising the value of ℓ in the OTM scheme has no consequences on the size of signatures, since the challenge e is computed from a short seed anyway (say 80 bits). Thus, even taking into account the slight increase of a that was suggested above, the scheme still beats GPS in terms of signature size, since it achieves bit size 320 whereas GPS stands over 400 bits.

On the other hand, we claim that, in terms of efficiency, the OTM scheme now has much worse performance, when aiming at a comparable security level 2^{80}. Indeed, the core operation that GPS performs on the fly, is the (regular) multiplication of the secret key s (160 bits) by the challenge e (80 bits). By the usual shift and add algorithm, this means 40 additions of a 160 bit integer. This computation sums up to about 200 8-bits multiplication or about 800 8-bit additions, if we are in a multiplication free context.

Let us now look at the core operation involved in OTM in more detail. This operation is the modular reduction of the random challenge e by the secret key s. With the new security parameters suggested above, e is 320 bits and s is 160 bits. We note that s and e must be chosen at random to maintain security, and therefore that they cannot be chosen of a certain specific form that could speed up modular reductions.

Hence, the efficiency of OTM is essentially the efficiency of general modular reductions of 320 bit numbers by 160 bit numbers. The modular reduction prim-

itive on large numbers has been extensively studied in [1], where the authors compare the performances of the three main reduction techniques: the so-called "classical" one, Barrett's algorithm and Montgomery's algorithm.

It turns out, both from their theoretical analysis and from their implementation on a 80386, that the reduction of a $(2k)$-bit number by a k-bit number is roughly comparable to the multiplication of 2 k-bit numbers, with an advantage for the multiplication (it costs between 420 and 480 8-bit multiplications, if we neglect the precalculation and postcalculation steps).

Thus, with the corrected security parameters, the core modular reduction needed by OTM is more than twice more expensive as the core multiplication needed by GPS.

3 A Signature Scheme Based on Dovetailing

In this section, we suggest to replace modular reduction by another novel operation, which we call *dovetailing*.

3.1 Dovetailing

Let s be an odd integer. Let (r, e) be a pair of integers, $r > s$, $r > e$. Dovetailing (r, e) with respect to s is adding to r a small multiple of s so to as to make the trailing bits of the result match with e. In mathematical terms, the operation might look intricate: setting $y = r + \lambda s$, one can choose λ as $(e - r)s^{-1} \bmod 2^k$, where k is the bit size of e. However, as will be seen in the sequel, it can be performed by a surprisingly efficient algorithm. We let $\mathcal{D}_s(r, e)$ be the result of dovetailing.

3.2 The New Scheme

We now introduce a scheme that uses dovetailing in place of modular reduction. The setting is essentially identical to OTM, except that we need to assume that the order s of g is odd.

The security analysis is similar to the (corrected) analysis of OTM: probability of forgery is of the order $1/2^\ell$ and statistical zero knowledge of a single round of identification is controlled by $2s/2^{a-k-\ell}$. Thus, we suggest to set $k = 160$, $a = 320$, $\ell = 80$.

The signature scheme is derived as usual. Note that, since the trailing bits of y match up with e, it is enough to output y, which means 320 bits for the suggested parameters.

It should also be noted that checking the size of the response, (e.g. checking that $y \stackrel{?}{<} 2^a + 2^{k+\ell}$) is not critical for the security of our scheme but is preferable for implementation reasons.

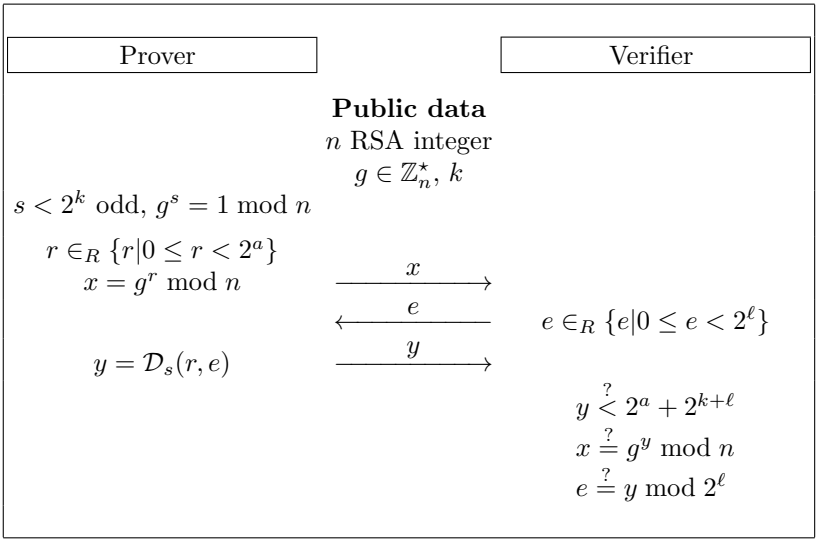

Fig. 3. An identification scheme based on dovetailing

3.3 Implementation

While general modular reductions are fairly complex algorithms, the special case of dovetailing allows for very efficient implementations.

Software implementation. We will first show that dovetailing can be implemented in software as fast as the primitive operation of GPS. In other words, dovetailing a (320-bit, 80-bit) pair with respect to a 160-bit integer can be implemented at least as efficiently as multiplying a 160-bit number by an 80-bit one.

We first start with the most natural and simpler dovetailing algorithm, described in algorithm 1. This algorithm requires on average 40 additions of a 160 bit number with a 320 bit number. If we neglect the few extra additions caused by the possible propagation of the carry in some pathological cases, the number of 8-bit additions is the same as in GPS.

Now, it is almost always as fast to substract an integer as to add it. We can take advantage of this property to speed up the computation time of the algorithm by optimizing on the fly the number of bits where r and e are equal. We assume in algorithm 2 that s is equal to 1 mod 4. The case where $s = 3 \bmod 4$ is similar.

It is fairly straightforward to compute the cost of this algorithm: when r and e match on a bit, there is simply a shift of one bit and when they don't, there is one addition or one subtraction which makes them match on two bits and a shift of two bits.

Algorithm 1 Simple dovetailing algorithm

INPUT: r, e, s
OUTPUT: $\mathcal{D}_s(r, e)$

```
i ← 0
while i < |e| do
  if e_i ≠ r_i then
    r ← r + s
  end if
  s ← s << 1
  i ← i + 1
end while
```

Hence, the average number of addition/subtraction of a 320 bit number and a 160 bit number needed to complete a run of the algorithm is $\frac{|e|}{3} \approx 27$, which means that approximately 540 additions on 8 bit numbers are needed, which has to be compared with the 800 additions in case of GPS. It is certainly possible to optimize GPS in a similar way by introducing substractions in the usual shift-and-add algorithm for multiplication. However, this will involve carries that dovetailing does not need, and result in a slightly less efficient implementation.

One can further optimize the scheme at the cost of creating a table of precomputed small multiples of s. To compute a multiple of s to add in the dovetailing algorithm, the table could be indexed according to the difference between the last d bits of e and r, taken modulo 2^d. Given this approach, d might be made larger than 2. For example, a table with $2^8 = 256$ entries would permit $d = 8$ and reduce the cost to about 200 additions on 8 bit numbers. We will not pursue this matter here as it is unclear whether the corresponding memory can be available in the setting of low cost smart cards that we envision.

Hardware implementation. We will now see that dovetailing can also be implemented fairly efficiently in hardware. Such an implementation would again be of a similar magnitude as the one for GPS.

We recall that the context of hardware implementation of these algorithms is simplicity and low cost, as they are reasonable candidates for performing digital signatures on memory cards. As a consequence, we considered the simplest and most compact designs.

The simplest way to implement a multiplication in hardware is by using the shift and add paradigm. Typically, one input is presented in bit parallel form while the other is in bit serial form. Each bit in the serial input "multiplies" the parallel input by either 0 or 1. The parallel input is held constant while each bit of the serial input is presented. Then, the result from each bit is added to an accumulated sum, which is shifted one bit before the next addition.

Now, when working on fairly large numbers, one will use a chain of unitary full-adders (say 8 bits), and might not want to wait until the carries fully propagate. This can be fairly easily solved by adding, at each cycle, the current value

Algorithm 2 Twobits dovetailing algorithm

INPUT: r, e, s
OUTPUT: $\mathcal{D}_s(r, e)$

```
i ← 0
while i < |e| do
  if e_i = r_i then
    s ← s << 1
    i ← i + 1
  else
    if i = |e| − 1 then
      r ← r + s
      i ← i + 1
    else
      t = (2r_{i+1} + r_i − 2e_{i+1} − e_i) mod 4
      if t = 1 then
        r ← r − s
      else
        r ← r + s
      end if
      s ← s << 2
      i ← i + 2
    end if
  end if
end while
```

of the result, the value of this round input *and* the carry of the previous cycle. When all the additions have been performed the remaining carry is propagated.

At first, it seems to be harder to do such an optimization with dovetailing because the current value of the result should be used in the input test. However, we remark that in the implementation of dovetailing described above, the equality tests of the loop can be performed *before* the addition is completed. As a matter of fact, while it is necessary to eventually complete the additions so that r yields the correct result, we just need the ith bit of r to be correct when performing the ith iteration of the loop.

Hence it seems possible to implement a hardware design following the general principle described below:

- The bits of r and e are compared to figure out whether to add 0 or s to r (easily done with a `MUX`).
- Additions are made with full adders of some given small size, say 8 bits.
- Every addition between r and s is performed in parallel on all the 8 bits full adder, *but* the carries between full adders are propagated only *once*. Remaining carries are reinserted in the full adders at the next round.
- s is shifted.

Because we propagate the carry only once every round, each addition between r and s only needs 2 clock cycles, and the carry propagation only need to be performed once after the main loop.

Going into more details, we note that it is in fact necessary to perform the extra propagation step only once every 8 cycles (if we are working with 8-bit full adders). So, at the price of a more complex clocking algorithm, it might be possible to perform dovetailing in hardware at virtually the same cost as the GPS multiplication.

4 Conclusion

We have shown an attack against the OTM signature scheme [8] presented at Financial Cryptography 2002. Although the scheme can be repaired in a straightforward manner by raising the parameters, this drastically degrades its performances in terms of computing time, making it less efficient than GPS. We have also designed another scheme, based on a novel operation called dovetailing, which can be implemented in a surprisingly efficient way. This scheme appears to achieve performances similar to OTM in terms of key size, and similar to GPS in terms of computing power requirements, thus yielding a strictly better scheme than both of them, by matching their respective strengths. We argue that both our scheme and GPS plausibly qualify as a candidate for execution on a memory card rather than a smart card. Still, our scheme is better in terms of signature size.

Acknowledgements. We wish to thank Marc Girault, Guillaume Poupard and Adi Shamir for nice discussions around the GPS scheme and its variants. We also thank the anonymous referees for their constructive remarks and suggestions. Finally, we wish to mention the respective contributions to the present work of the two authors: the first named author is responsible for the cryptanalysis, while the second named author has found the new design and the corresponding implementations.

References

1. A. Bosselaers and R. Govaerts and J. Vandewalle Comparison of three modular reduction functions *Proceedings of Crypto 93*, Lecture Notes in Computer Science 773, pages 175–186, 1994.
2. U. Feige and A. Fiat and A. Shamir. Zero-Knowledge Proofs of Identity. *J. Cryptology*, 1, 1988, 77–94.
3. S. Even, O. Goldreich and S. Micali. Online/Offline Digital Signatures. In *Journal of Cryptology*, 9, 1996, 35–67.
4. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings of Crypto 86*, Lecture Notes in Computer Science 263, 1987, 181–187.
5. M. Girault. Self-certified Public Keys. In *Proceedings of Crypto 91*, Lecture Notes in Computer Science 547, Springer-Verlag, 1992, 490–497.
6. L.S. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory. In *Proceedings of Eurocrypt 88*, Lecture Notes in Computer Science 339, Springer-Verlag, 1989, 123–128.

7. N. Lygeros, M. Mizony, P. Zimmermann, A new ECM record with 54 digits, `http://www.desargues.univ-lyon1.fr/home/lygeros/Mensa/ecm54.html`
8. T. Okamoto, M. Tada and A. Miyaji. An Improved Fast Signature Scheme without on-line Multiplication. In *Pre-proceedings of Financial Cryptography 02*.
9. J. Pollard. Monte Carlo methods for index computation mod p, *Math. Comp.*, 32, 1978, 918–924.
10. G. Poupard and J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. In *Proceedings of Eurocrypt 96*, Lecture Notes in Computer Science 1403, Springer-Verlag, 1998, 422–436.
11. G. Poupard and J. Stern. On the fly Signatures based on Factoring. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, ACM Press, 1999, 48–57.
12. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Proceedings of Crypto '89*, Lecture Notes in Computer Science 435, Springer-Verlag, 1990, 235–251,
13. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4, 1991, 161–174.
14. A. Shamir and Y. Tauman. Improved Online/Offline Signatures Schemes. In Crypto 2001, Lecture Notes in Computer Science 2139, Springer-Verlag 2001, 355–367.

"Man in the Middle" Attacks on Bluetooth

Dennis Kügler

Federal Office for Information Security
Godesberger Allee 185-189
53133 Bonn, Germany
Dennis.Kuegler@bsi.bund.de

Abstract. Bluetooth is a short range wireless communication technology that has been designed to eliminate wires between both stationary and mobile devices. As wireless communication is much more vulnerable to attacks, Bluetooth provides authentication and encryption on the link level. However, the employed frequency hopping spread spectrum method can be exploited for sophisticated man in the middle attacks. While the built-in point-to-point encryption could have offered some protection against man in the middle attacks, a flaw in the specification nullifies this countermeasure.

1 Introduction

The Bluetooth specification [SIGa] is an industrial standard for short range wireless communication between both stationary and mobile devices. As wireless communication is vulnerable to several attacks, Bluetooth employs strong (symmetric) cryptography to enable secure and private communication between Bluetooth devices. However, several weaknesses have already been found in Bluetooth's security architecture. Most of them are discussed in a paper by Jakobsson and Wetzel [JW01] that presents a passive PIN guessing attack, a mechanism to attack privacy by tracking Bluetooth devices, a man in the middle attack, and some flaws in the used stream cipher E0. A closer examination of E0 has been done by Fluhrer and Lucks [FL01]: Although the Bluetooth specification allows key lengths in the range from 1-16 bytes (8-128 bits), they show that the strength of the cipher will not exceed 84 bits or 73 bits depending on the number of the key stream bits that are known to the attacker (132 bits or 2^{43} bits, respectively).

Most of the already known attacks on Bluetooth can be circumvented or fixed more or less easily. A paper by Gehrmann and Nyberg [GN01] presents some enhancements to Bluetooth security that are currently under review by the Bluetooth SIG. Among other things they discuss a solution to the PIN guessing attack based on encrypted key exchange [BM92] and some countermeasures to the tracking attack, which unfortunately still leaks the slave's device address at paging.

One well-known problem has been overlooked so far: Bluetooth does not provide a per packet cryptographic integrity check that prevents data from being

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 149–161, 2003.
© Springer-Verlag Berlin Heidelberg 2003

manipulated while transmitted. However, without being able to intercept transmitted data, i.e. with a man in the middle attack, exploiting this security hole is difficult. As the man in the middle attack presented by Jakobsson and Wetzel can be fixed relatively easy by configuring devices accordingly, we believe that this security hole will be disregarded in upcoming versions of Bluetooth. To emphasize the strong need for appropriate countermeasures against man in the middle attacks, we present a novel man in the middle attack and an improved version of the attack of Jakobsson and Wetzel.

The remainder is structured as follows: Section 2 explains the connection establishment procedures of Bluetooth. Those procedures are the foundation for our man in the middle attack presented in Section 3: We show how an attacker is able to interpose between the devices establishing a connection. We also discuss how link encryption affects our attack. Then in Section 4 we compare our attack to the attack of Jakobsson and Wetzel but also improve their attack. We summarize the exploited properties and discuss some countermeasures in Section 5. Finally, we conclude our paper in Section 6.

2 Connection Establishment

For the reader's convenience, we will start with a very short introduction to Bluetooth: A network of Bluetooth devices is called *piconet* and consists of a *master* and up to 7 active *slaves*. When two devices establish a connection, a small piconet consisting of both devices is formed. The initiating device becomes the master of this piconet and the other device becomes the slave. However, if both devices agree to switch roles, the initiating device becomes a slave of the other device, who may already have formed a piconet.

Every Bluetooth device is associated with a unique 48 bit device address (BD_ADDR) and a 28 bit clock (CLK) running at 3.2 kHz and thus wrapping every 23.3 hours. A time devision duplex communication scheme is used, where the slave is synchronized to the clock setting of the master. The master starts transmissions in even slots ($CLK_0 = 0$ and $CLK_1 = 0$) and the previously addressed slave may respond in odd slots ($CLK_0 = 0$ and $CLK_1 = 1$). To avoid interference of several piconets operating in the same area, a frequency hopping mechanism is employed.

The following protocols are used to establish a connection:

1. **Inquiry:** The inquiry protocol is used to discover unknown devices in the area. The discoverability of a device can be restricted by the user:
   a) A limited discoverable device is only temporarily discoverable.
   b) A non-discoverable device will never respond to an inquiry message.
2. **Paging:** The paging protocol is used to establish a connection to an already discovered device. The connectability of a device can be restricted by the user: A non-connectable device will never scan for page requests.

3. **Pairing:** The pairing protocol is used at the first interconnection of two devices to setup link keys for subsequently establishing authenticated connections. The pairability of a device can be restricted by the user: Pairing with a non-pairable device is not possible.

In the following, we will explain the frequency hopping mechanism and the paging procedure in detail, as they are the foundation for our man in the middle attack.

2.1 Frequency Hopping

Frequency hopping is a spread spectrum method that is used to make a narrow-band signal more robust against noise and interferences. The available broadband spectrum is divided into several narrow-band spectrums. The narrow-band signal is distributed to the broadband spectrum by hopping between those narrow-band spectrums while transmitting or receiving messages. The sequence of the used narrow-band spectrums is called the *hopping sequence*. The transmitter and the receiver must be synchronized to use the same hopping sequence and the same offset in this sequence, i.e. both devices must use the same frequency at the same time.

Bluetooth divides the broadband spectrum at 2.4 GHz into 79 narrow-band spectrums. Three types of hopping sequences are used:

Inquiry Hopping Sequence: This sequence consists of 32 frequencies, has a period of length 32 and is always determined by the general inquiry access code (GIAC) and the clock of the discovering device.

Page Hopping Sequence: This sequence also consists of 32 frequencies and has a period of length 32 but is determined by the device address and the clock of the slave.

Channel Hopping Sequence: This sequence consists of all 79 frequencies and is determined by the device address and the clock of the master. This sequence has a very long period, which does not show repetitive patterns over a short time interval.

2.2 Paging

To establish a connection, master and slave first have to be synchronized to the same page hopping sequence. Thus, the master has to know the device address and the clock setting of the slave. If both devices are to be connected for the first time, the master uses an inquiry to learn the parameters of the slave.

But as clocks always drift after a while, the slave's clock may not be *exactly* known to the master. To compensate drifts the following strategy is used: The master first guesses the clock setting of the slave from the last connection, calculates the expected offset k in the page hopping sequence and rapidly sends out a number of page requests (on A- and B-trains, see below). The slave however only hops to the next frequency every 1,28 s and scans for page requests for a

certain time. After each sent page request the master listens for a slave response. This procedure is repeated until either the master receives a slave response or a timeout occurs.

A-Train: The master sends out page requests on the A-train, which consists of the frequencies $f(k-8), f(k-7), \ldots, f(k), \ldots, f(k+7)$. The A-train is repeated N times. If the slave does not respond, the B-train is tried.

B-Train: The B-train consists of the remaining frequencies $f(k-16), f(k-15), \ldots, f(k-9), f(k+8), \ldots, f(k+15)$. The B-train is also repeated N times. If the slave still does not respond, the A-train is tried again etc.

As the page requests are transmitted in both send and receive slots, sending a complete train takes only 10 ms. As every connectable device enters the scan sub-state either every 1,28 s or every 2,56 s, it is necessary to repeat every train for either $N = 128$ or $N = 256$ times, depending on the scanning behavior of the slave (which should be known to the master from the inquiry). Thus, the master should receive a page response at least after 2,56 s or 5,12 s, respectively. Altogether, the paging procedure consists of the following steps:

1. **Page Request:** The master sends out a number of ID packets on different frequencies containing the *device access code* (DAC) of the slave. The DAC is calculated from the lower 24 bits of the 48 bit device address of the slave.
2. **Slave Response:** The slave eventually scans for ID packets containing its own device access code. After reception of a matching ID packet the slave responds with the same ID packet.
3. **Master Response:** When the master receives the slave response, both devices are synchronized and the master responds with an FHS packet containing its own device address and its actual clock setting.
4. **2nd Slave Response:** The slave responds again with an ID packet. After the master has received the second slave response, the communication is switched to the channel hopping sequence.

The slave may request to switch roles directly after paging. If the master agrees, he will become the slave of the paged device, who may already have established an piconet. This *master-slave switch* may also be requested by both master and slave at any time after the connection is established.

3 Cutting the Wire

In this section we will show that Bluetooth fails to be a secure replacement for wire based communication. We present a novel man in the middle attack that is able to intercept regular Bluetooth traffic.

3.1 Attacks Induced by Paging

It is often said that frequency hopping increases security, as synchronization to the hopping sequence is not easy. We will show that this is not the case. Actually,

frequency hopping decreases security, as its properties can be exploited for a man in the middle attack.

Our attack makes use of the paging procedure. As the master does not know when the paged device starts the page scan procedure, the master has to sequentially send a number of page requests until the slave responds. The attacker proceeds as follows:

1. The attacker responds faster than the slave to the page request (how this can be achieved by the attacker is described below). Thus, the master establishes a connection with the attacker instead of the slave. After the attacker is connected to the master, the channel hopping sequence of the master will be used for communication.
2. Now the attacker restarts the paging procedure with the slave. After the slave has returned the slave response, the attacker will respond with a FHS packet that contains a clock setting that is different from the master's clock setting. Thus, master and slave will use the same channel hopping sequence, but a different offset in this sequence.
3. If the slave requests a role switch, the current channel hopping sequence is used until the switch is finally agreed and an FHS packet is sent from the new master to the old master. As the attacker is able to alter the FHS packet, both devices will use the new channel hopping sequence, but still a different offset.

Thus, as master and slave use the same channel hopping sequence, but always a different offset in this sequence, it is guaranteed that the attacker can intercept every communication between both devices. Next, we show how it can be achieved that the master establishes a connection with the attacker instead of the slave. We present two methods that prevent the slave from responding to a page request:

Jamming. In most cases the slave will not immediately receive the page request. Thus, an attacker scanning all 32 page hopping frequencies in parallel will be able to respond to the first page message and set up the communication with the master. There is however a small chance that the slave will respond to the first page request. In this case both the attacker and the slave send out the slave response *nearly* in parallel. In this case the synchronization will fail completely as the channel is jammed. The master will send out the next page request on the next frequency to which only the attacker is able to respond as the slave changes its scanning frequency only every 1,28 s or 2,56 s (see also Figure 1). If the slave coincidentally also changes its scanning frequency, an additional "jam step" is required.

Busy Waiting. Alternatively, the attacker starts establishing a connection to the slave before the master does. However, the attacker interrupts the connection establishment before sending the master response (the FHS packet). Then the

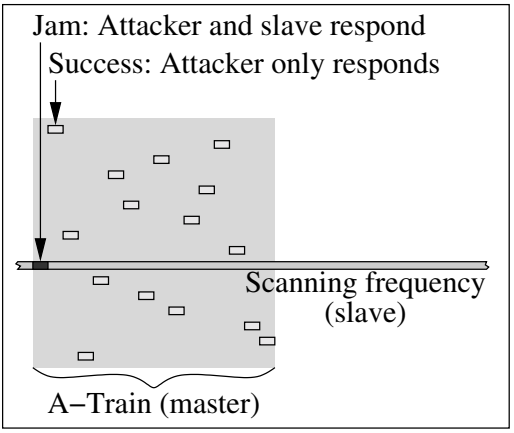

Fig. 1. The man in the middle attack with one jam step.

connection is "half open" and the slave continuously scans for a FHS packet until a timeout occurs. After this timeout, the slave once again scans for a certain time for an ID packet. If the slave does not receive an ID packet, the slave returns to normal operation [SIGa, page 103].

The attacker can exploit this behavior and hold the slave most of the time in this "half open" state. As the master sends ID packets, but the slave scans for FHS packets, only the attacker is able to respond to the page request. After the attacker has received the FHS packet from the master, he can immediately send a FHS packet to the slave. Thus, the attacker can simultaneously establish the connection to master and slave.

3.2 Attacking an Encrypted Link

Bluetooth uses a stream cipher for optional encryption. The cipher is rekeyed with a new IV for every transmitted or received packet. The IV is internally computed in both devices from the device address and the clock of the master.

If encryption is turned on, the integrity of the encrypted payload is protected by a CRC that is appended to the payload before encryption. It is a well known result that such an integrity check does not protect against intended manipulation of the ciphertext. It is possible to toggle single bits in the ciphertext, which will directly affect the corresponding bit in the plaintext. In [BGW01] it is shown how to manipulate an intercepted ciphertext with encrypted CRC, so that the resulting plaintext is correct (according to the CRC) but changed in a way known to the attacker. While the attacker still is not able to eavesdrop on the encrypted communication, he may manipulate every intercepted packet, before forwarding it to the recipient. Together with full or partial knowledge of the plaintext (e.g. IP headers), this is a powerful attack.

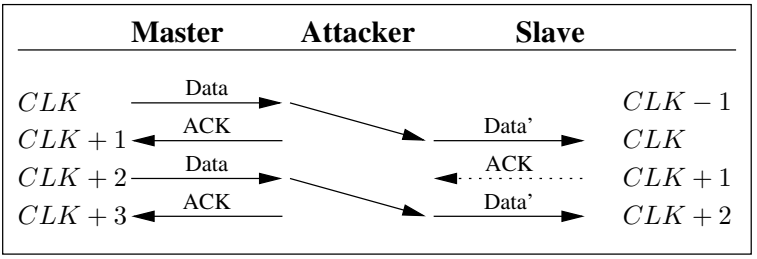

Fig. 2. Attacking unidirectional communication.

Unidirectional Communication. If only one device sends data while the other device acknowledges reception, the attack is relatively simple: As Bluetooth offers reliable transport channels, acknowledge messages are sent on the link level. Unfortunately, the acknowledge flag is included in the packet header [SIGa, page 51], but only the packet payload is encrypted [SIGa, page 158ff.].

To exploit this, we assume that data is transmitted from master to slave, and that the clock setting of the slave is at least one tick behind the master. For every intercepted data packet, the attacker proceeds as shown in Figure 2:

1. The attacker responds to the master with an unencrypted ACK packet to acknowledge the reception of the data packet.
2. The attacker manipulates the packet and forwards it to the slave in the corresponding slot.
3. The attacker discards ACK packets received from the slave.

Transmission errors are a possible problem. If the slave does not receive a correct data packet, it will respond with a NAK packet, but the attacker is unable to resend the data packet encrypted with an updated IV. Therefore, the attacker should always responds to the master with a NAK packet until an ACK packet is received from the slave. A drawback of this method is the bandwith reduction as the master always has to send every data packet at least twice.

Bidirectional Communication. If both devices send and receive data packets, the man in the middle attack should fail: To succeed with our attack, it is crucial that both devices use the same IV to en- and decrypt packets. However, the attacker lets the victim's devices use different offsets in the same hopping sequence, which in turn depends on having the slave using a clock setting that is different from the master's clock setting. As master and slave use a different IV to en- and decrypt packets, our attack should fail.

Surprisingly, our attack does not fail. The IV only depends on CLK_{1-26} [SIGa, pages 162ff.], while the channel hopping sequence is determined by the full clock CLK_{1-27} [SIGa, pages 128ff.]. Thus, we are able to flip the bit CLK_{27}

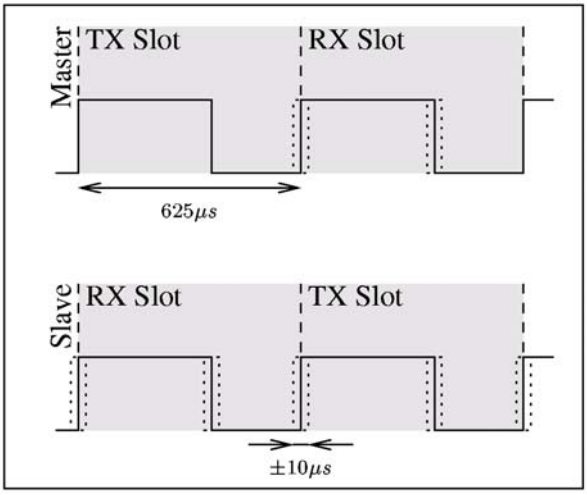

Fig. 3. The uncertainty window.

in the FHS packet at connection establishment. Then both clocks drift approximately 11.65 hours so that master and slave use very different offsets in the channel hopping sequence but the same IV.

Due to their synchronized clocks (at least for the lower bits), master and slave are tight coupled and the attacker has not much time to manipulate the ciphertext. In fact the attacker has to retransmit every single bit received. A slight delay is however possible as every slot has an uncertainty window of $\pm 10\mu$s as shown in Figure 3.

4 Improving the Jakobsson-Wetzel Attack

Our man in the middle attack exploits the connection establishment procedures to intercept a regular Bluetooth connection. Another man in the middle attack was previously presented by Jakobsson and Wetzel [JW01]. In contrast to our attack, their attack requires both victim devices to be connectable, which is not always the case [SIGb]. If one device is non-connectable, the attack fails. The man in the middle attack of Jakobsson and Wetzel is basically performed as follows:

> The attacker simultaneously establishes a connection to both devices, posing to be the other device. Then each device believes that it talks to the other device who also has initiated the communication. Therefore, both devices are slaves and will use the hopping sequence of the other device who is claimed to be the master. Thus both devices use different channel hopping sequences and will not hear each other.

Jakobsson and Wetzel also mentioned the option to request role switching while establishing the connection to both devices. Thus, the attacked devices will become both masters, which has the advantage that the communication channel is not jammed if one device is already the master of another piconet. However, this option is not possible as, according to the Bluetooth specification [SIGa, pages 1044ff.], only the slave can initiate a role switch at connection establishment. Thus, both devices would have to request role switching while establishing the connection to the attacker. But even in this unlikely case the attacker is still the master of the piconet until the roles are finally switched.

4.1 Attacking an Encrypted Link

The attack also fails if one device requests encryption. As both devices are slaves and believe that the other device is the master, both devices use a different device address to calculate the IV. Thus, the attack will fail if the link key is unavailable to the attacker. Therefore, Jakobsson and Wetzel suggest that the attacker may pretend to have lost the link key. In this case a new link key will be negotiated. If a weak PIN is used to setup the new link key, the attacker can calculate the link key from the eavesdropped transcripts ("PIN guessing attack").

However, the success of this attack depends on both devices configured to be pairable and on the user entering a weak PIN to setup the new link key. Furthermore, if the user has to manually enter a PIN in at least one device, this renegotiation may alert the user.

4.2 Improving the Attack

Unlike stated in the original attack, it is not necessary that the victim devices are either both slaves or both masters to prevent jamming. Similar to our attack, we now let both devices use the same channel hopping sequence but a different offset in the sequence. Therefore, the attacker connects both devices and switches roles with one device after the connection is established. Let A and B be the victim's devices where A is usually the master. The attacker proceeds as shown in Figure 4:

1. The attacker establishes a connection to both devices posing to be the other device. As the attacker is the master and the victim's devices are both slaves the attacker chooses the clock setting $CLK = CLK_A \oplus 2^{27}$ to be used.
2. After the connections are established both devices use the same offset in different channel hopping sequences. Now the attacker switches roles with device A (note that A may already have requested a role switch at connection establishment). Afterwards both devices use the same channel hopping sequence, but different clocks (CLK_A respective $CLK_A \oplus 2^{27}$).
3. The attacker proceeds as described in section 3.2.

To succeed with this attack it is still important that both devices are connectable and are not the master of another piconet. In addition device A must not be

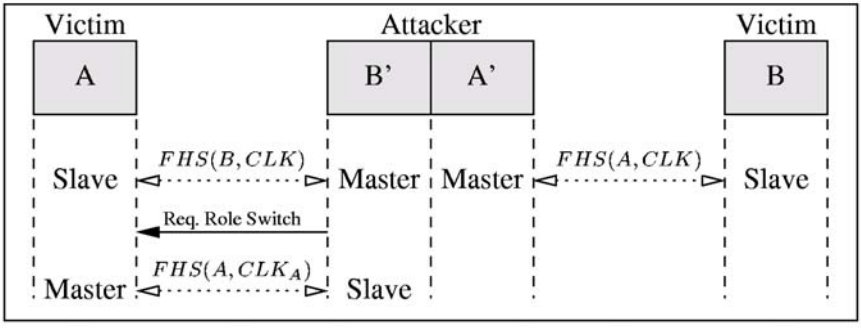

Fig. 4. The improved Jakobsson-Wetzel attack.

configured to request encryption while the connection is established [SIGb, page 37], otherwise the attacker cannot request role switching any more. However, if A also requests role switching at connection establishment, the attack is still successful, as this is done before encryption is started [SIGa, pages 1044ff.].

5 Preventing Man in the Middle Attacks

We have shown that Bluetooth's frequency hopping mechanism can be exploited for man in the middle attacks: it is possible to have two devices establish a connection with unsynchronized clocks. As a consequence, both devices use the same hopping sequence but a different offset in that sequence. Both devices thus communicate on different frequencies and will not receive the messages sent to each other. This enables a man in the middle to intercept, manipulate and forward the transmitted messages.

Bluetooth uses a clever mechanism to counteract such attacks: The master's clock setting is used for both, as offset in the frequency hopping sequence and as IV for encryption. Assuming that strong encryption is turned on *automatically* directly after connection establishment, this could have prevented man in the middle attacks to a certain extent.

5.1 Exploited Properties

Unfortunately, encryption is not turned on per default. But even if encryption were turned on, the man in the middle attack would have been possible due to the following additional flaws:

– Bluetooth encryption uses a per packet IV that is derived from the master clock. However, the most significant bit of the clock is not taken into account. As the offset in the page hopping sequence is derived from the full clock, this bit can be flipped to bring both devices out of synchronization with respect to the hopping sequence.

- Bluetooth offers reliable communication. This is implemented by including an acknowledge flag in the packet header. However, the packet header is never encrypted. Thus, a man in the middle can acknowledge packets which are forwarded later.
- Bluetooth offers an uncertainty window around every slot, to allow minor delays in transmissions. As this uncertainty window is quite large, an attacker can relay single bits between both devices.

An attacker who wants to exploit those flaws still has to face the problem that he has to relay single bits. Thus, for every received bit the attacker immediately has to decide whether to forward this bit unaltered or flipped. However, some of the attacks to intercept IP traffic presented in [BGW01] require the attacker to have knowledge of the whole encrypted packet before relaying it. Therefore, alternative strategies can be used:

- The attacker first receives a packet without relaying it and responds with a negative acknowledge. Then the master will retransmit the packet later. In the meantime, the attacker has enough time to analyze which bits are to be flipped in the retransmitted packet.
- When the attacker relays a manipulated packet to the slave the attacker observes the reaction of the slave. The attacker responds with negative acknowledges until the slave's reaction is convenient for the attacker.

5.2 Circumventing Encryption

The best strategy for the attacker is to circumvent encryption completely. As encryption is always optional and needs to be negotiated between both devices, a man in the middle can either reject encryption requests without forwarding them to the other device or setup encryption with a very short cipher key. Bluetooth allows cipher keys down to 8 bits. If both devices accept a short cipher key, encryption can be broken quite easily by an attacker. Unfortunately, the applications are not aware of the length of the negotiated cipher key.

Encryption can also be circumvented, if the master device decides to use broadcast encryption. Therefore, it transfers a temporary link key K_{master} to every participant of the piconet before encryption is started [SIGa, page 201]. If the attacker is a legitimate member of the piconet, using broadcast encryption is quite similar to using no encryption.

5.3 Fixing the Problems

The security mechanisms integrated in Bluetooth are not adequate. However, man in the middle attacks become much more difficult, if strong point-to-point encryption is used and it is additionally guaranteed that both devices use the same offset in the same hopping sequence. In this case the level of security provided is perhaps comparable to wire based communication. Man in the middle

attacks seem to be only possible if both devices are out of range and are thus unable to (directly) communicate with each other.

To guarantee that the clocks of two devices are indeed synchronized, either the full clock should be used to derive the IV, or the integrity of the clock should be verified by e.g. using the master clock as additional input in the authentication protocol.

To prevent all attacks, including a cryptographic integrity check based on the link key in every packet is inevitable. However, it should be noted that this still doesn't provide end-to-end security like IPSec or SSL/TLS.

6 Conclusion

While Bluetooth has several nice features, it fails to be a secure replacement for wires. We have shown that Bluetooth is susceptible to man in the middle attacks, independent of the employed security mechanisms. Therefore, our attack is superior to the man in the middle attack presented by Jakobsson and Wetzel, which depends on insecurely configured devices that must at least be both connectable.

As our attack is based on bringing devices out of synchronization, while Bluetooth's encryption mechanism requires tight synchronization, our attack could have been prevented to some extent by turning on strong point-to-point encryption. Unfortunately, the specification contains a flaw that nullifies the synchronization requirement. We also expect that the adaptive frequency hopping mechanism that will be available in Bluetooth 1.2 can be exploited even more for man in the middle attacks. Therefore, we hope that upcoming versions of Bluetooth integrate an appropriate cryptographic integrity check in every packet.

Acknowledgements. The author would like to thank the anonymous referees for valueable comments especially for being pointed to the paper of Gehrmann and Nyberg.

References

[BGW01] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: The insecurity of 802.11. In *7th Annual International Conference on Mobile Computing and Networking*. ACM Press, 2001.

[BM92] Steven M. Bellovin and Michael Merrit. Encrypted key exchange: Password-based protocols against dictionary attacks. In *IEEE Symposium on Research in Security and Privacy*, pages 72–84. IEEE Computer Society Press, 1992.

[FL01] Scott R. Fluhrer and Stefan Lucks. Analysis of the E0 encryption system. In *Selected Areas in Cryptography – SAC 2001*, Lecture Notes in Computer Science 2259, pages 38–48. Springer-Verlag, 2001.

[GN01] Christian Gehrmann and Kaisa Nyberg. Enhancements to Bluetooth baseband security. In *Nordic Workshop on Secure IT-Systems – NordSec 2001*, pages 39–53. Proceedings, 2001.

[JW01] Markus Jakobsson and Susanne Wetzel. Security weaknesses in Bluetooth. In *Progress in Cryptography – CT-RSA 2001*, Lecture Notes in Computer Science 2020, pages 176–191. Springer-Verlag, 2001.

[SIGa] Bluetooth SIG. Specification of the Bluetooth system: Core, version 1.1. http://www.bluetooth.org.

[SIGb] Bluetooth SIG. Specification of the Bluetooth system: Profiles, version 1.1. http://www.bluetooth.org.

Fault Based Cryptanalysis of the Advanced Encryption Standard (AES)

Johannes Blömer[1] and Jean-Pierre Seifert[2]

[1] University of Paderborn
D-33095 Paderborn, Germany
bloemer@upb.de
[2] Infineon Technologies, Secure Mobile Solutions
D-81609 Munich, Germany
Jean-Pierre.Seifert@infineon.com

Abstract. In this paper we describe several fault attacks on the Advanced Encryption Standard (AES). First, using optical/eddy current fault induction attacks as recently publicly presented by Skorobogatov, Anderson and Quisquater, Samyde [SA,QS], we present an implementation independent fault attack on AES. This attack is able to determine the complete 128-bit secret key of a sealed tamper-proof smartcard by generating 128 faulty cipher texts. Second, we present several implementation-dependent fault attacks on AES. These attacks rely on the observation that due to the AES's known timing analysis vulnerability (as pointed out by Koeune and Quisquater [KQ]), any implementation of the AES must ensure a data independent timing behavior for the so called AES's `xtime` operation. We present fault attacks on AES based on various timing analysis resistant implementations of the `xtime`-operation. Our strongest attack in this direction uses a very liberal fault model and requires only 256 faulty encryptions to determine a 128-bit key.

Keywords: AES, Fault Attacks, Implementation Issues, Secure Banking, Smart Cards.

1 Introduction

Recently, Rijndael has been approved by the NIST as the Advanced Encryption Standard (AES). This makes the AES compulsory and binding on Federal agencies for the protection of sensitive unclassified information. Thus, within the near future AES will replace DES as the worldwide standard in symmetric key encryption. Especially for financial applications this replacement of the DES by the AES will be very important. Not surprisingly, a lot of research so far focused on the mathematical security of AES. Likewise, the security of AES against side-channel attacks like timing analysis and power analysis, has also been examined, cf. [AG,BS99,CJRR,DR1,KQ,KWMK,Me,YT]. On the other hand, no fault-based cryptanalysis of AES has been reported so far. This is surprising as

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 162–181, 2003.
© Springer-Verlag Berlin Heidelberg 2003

the frauds with smartcards by inducing faults are real, cf. [A,AK1,AK2], whereas no frauds via Timing or Power Analysis attacks have been reported so far.

In this paper we describe several methods for a fault based cryptanalysis of AES. We present an implementation independent attack as well as attacks on several implementations of AES aimed at making AES timing analysis secure.

Several different fault models are used in our attacks. The first attack is implementation independent but uses the most restrictive fault model. It is aimed at the first transformation during an encryption with AES, the so called `AddRoundKey`. In this attack we use a rather strong but seemingly realistic fault model. We assume, that an attacker can set a specific memory bit to a fixed value, e.g., to the value 0. Moreover, we assume that the attacker can do this at a precise time of his choice. The practicality of this model has recently been demonstrated in [SA,QS]. Using this model we show that an attacker can determine the complete 128-bit secret key by computing 128 encryptions inducing a single fault each time. We also show that the fault model can be relaxed in two ways. First, an attacker need not be able to set a fixed memory bit with probability 1 to the value 0, say. Instead, it is sufficient that the value of a certain a bit will be changed with slightly higher probability from 1 to 0 than from 0 to 1. Second, we show that the attacker need not be able to change the bit at a precise point in time. Hence, to fend off the attack it is not sufficient to equip a cryptographic device with a randomized timing behavior. We note that our implementation independent attack can also be mounted against other symmetric encryption ciphers like IDEA, SAFER, and Blowfish. In fact, like AES, these ciphers have an initial key addition. This alone renders them immediately susceptible to our implementation independent attack. We like to point out that our basic attack scenario is similar to an attack scenario described in [BS97]. However, our attack scenario is much simpler to realize and especially more practically oriented towards real threats.

Our second class of attacks are implementation dependent. More precisely, we consider several different implementations for the so called `xtime` operation that is performed during the transformation `MixColumn`. As it is well-known, AES is susceptible to timing attacks if `xtime` is not implemented appropriately. We consider various timing attack resistant implementations and show that all of them lead to a simple fault based cryptanalysis of AES. The various implementations we consider vary from software solutions to hardware based implementations. Definitely, we do not consider all implementations of `xtime` currently in use. However, we are confident that most if not all implementations are susceptible to the attacks described in this paper. The fault models we use in the attacks range from very liberal models (arbitrary fault at a specific location) to more restricted models, like the one we use in the implementation independent attack. Our strongest result shows that the most obvious timing attack resistant implementation of `xtime` opens the door for a simple fault based attack on AES. With this attack the complete secret key can be determined encrypting roughly 256 plaintexts. In each encryption an attacker must induce an *arbitrary* fault at a specific memory byte. Using the methods we describe for the implementation

independent attack, one can argue, that it is not necessary for the attacker to induce the fault at a specific time. We like to point out that our implementation dependent attacks use a technique introduced by [KQ] and were inspired by ideas in [YJ].

In all our attacks we assume, that from the behavior of a cryptographic device we can deduce whether a fault leading to a wrong ciphertext has actually occurred during an encryption. If a fault occurs, the device may simply output a wrong ciphertext. However, an attacker can compute the correct ciphertext by rerunning the device without inducing a fault, thereby detecting whether a fault occured during the original encryption. The device may also check its own result, or may even check intermediate results and answer with an error warning in case a fault has been detected (see [KWMK]). This clearly tells an attacker that a fault has occurred. Thirdly, the card may check whether a fault has occurred, and if so, recompute the encryption. However, in this case by measuring the time before the device outputs the ciphertext, an attacker can tell whether a fault occurred. To simplify the presentation, we assume that a cryptographic device will answer with the output `reset`, whenever a fault caused a wrong ciphertext.

From the results in this paper, it follows that it is absolutely necessary to incorporate both hardware and software means into AES realizations to guard against fault attacks. Of course, some modern high-end crypto smartcards are protected by various sophisticated hardware mechanisms to detect any intrusion attempt to their system behavior. However, as techniques to induce faults into the computations of smartcards become more sophisticated, it is mandatory to use also various software mechanisms to fend off fault attacks. We like to close with an advice due to Kaliski and Robshaw [KR] from RSA Laboratories that *good engineering practices in the design of secure hardware are essential*, we only would like to add, that the same applies to secure software.

The paper is organized as follows. We first describe the AES algorithm. In the next section we briefly turn to the physics of realizing faults during computations. In particular, we sketch and characterize the physical attacks used by us in later sections. The following section then describes our general independent fault attack on AES. In this section we also describe the two aforementioned relaxations concerning a probabilistic fault model and a loose time control on the induced error. The next section starts with the explanation of the basic idea for all of our implementation dependent fault attacks. Then, we will continue to describe incrementally more and more sophisticated fault attacks. Eventually, we give some hints on conceivable countermeasures to defeat our fault attacks on the AES.

2 Preliminaries

2.1 Description of the Advanced Encryption Standard

In this section we briefly describe the Advanced Encryption Standard (AES). For a more detailed description we refer to [DR2].

AES encrypts plaintexts consisting of lb bytes, where lb = 16, 24, or 32. The plaintext is organized as a $(4 \times \mathtt{Nb})$ array (a_{ij}), $0 \le i < 4, 0 \le j < \mathtt{Nb} - 1$, where $\mathtt{Nb} = 4, 6, 8$, depending on the value of lb. The n-th byte of the plaintext is stored in byte $a_{i,j}$ with $i = n \mod 4$, $j = \lfloor \frac{n}{4} \rfloor$.

AES uses a secret key, called *cipher key*, consisting of lk bytes, where lk = 16, 24, or 32. Any combination of values lb and lk is allowed. The cipher key is organized in a $4 \times \mathtt{Nk}$ array (k_{ij}), $0 \le i < 4, 0 \le j \le \mathtt{Nk} - 1$, where $\mathtt{Nk} = 4, 6, 8$, depending on the value of lk. The n-th key byte is stored in byte k_{ij} with $i = n \mod 4$, $j = \lfloor \frac{n}{4} \rfloor$.

The AES encryption process is composed of *rounds*. Except for the last round, each round consists of four transformations called ByteSub, ShiftRow, MixColumn, and AddRoundKey. In the last round the transformation MixColumn is omitted. The four transformations operate on intermediate results, called *states*. A state is a $4 \times \mathtt{Nb}$ array (a_{ij}) of bytes. Initially, the state is given by the plaintext to be encrypted. The number of rounds Nr is 10, 12, or 14, depending on $\max\{\mathtt{Nb}, \mathtt{Nk}\}$. In addition to the transformations performed in the Nr rounds there is an AddRoundKey applied to the plaintext prior to the first round. We call this the *initial* AddRoundKey.

Next, we are going to describe the transformations used in the AES encryption process. We begin with AddRoundKey.

The transformation AddRoundKey. The input to the transformation AddRoundKey is a state (a_{ij}), $0 \le i < 4, 0 \le j < \mathtt{Nb}$, and a *round key*, which is an array of bytes $(\mathtt{rk}_{ij})$, $0 \le i < 4, 0 \le j < \mathtt{Nb}$. The output of AddRoundKey is the state $(b_{ij}), 0 \le i < 4, 0 \le j < \mathtt{Nb}$, where

$$b_{ij} = a_{ij} \oplus \mathtt{rk}_{ij}.$$

The round keys are obtained from the cipher key by expanding the cipher key array (k_{ij}) into an array (k_{ij}), $0 \le i < 4, 0 \le j \le \mathtt{Nr} \cdot \mathtt{Nb}$, called the *expanded key*. The exact procedure by which the expanded key is obtained from the cipher key is of no importance for the attacks described in this paper. The round key for the initial application of AddRoundKey is given by the first Nb columns of the expanded key. Hence the round key for the first round is given by the cipher key itself. In general, the round key for the application of AddRoundKey in the m-th round of AES is given by columns $m\mathtt{Nb}, \ldots, (m+1)\mathtt{Nb} - 1$ of the expanded key, $1 \le m \le \mathtt{Nr}$.

The transformation ByteSub. Given a state (a_{ij}), $0 \le i < 4, 0 \le j < \mathtt{Nb}$, the transformation ByteSub applies an invertible function $S : \{0,1\}^8 \to \{0,1\}^8$ to each state byte a_{ij} separately. The exact nature of S is of no relevance for the attacks described later. We just mention that S is non-linear, and in fact, it is the only non-linear part of the AES encryption process. In practice, S is often realized by a substitution table or *S-box*.

The transformation ShiftRow. The transformation ShiftRow cyclically shifts each row of a state (a_{ij}) separately to the left. Row 0 is not shifted. Rows 1, 2, 3

are shifted by C_1, C_2, C_3 bytes, respectively, where the values of the C_i depend on `Nb`.

The transformation `MixColumn`. The transformation `MixColumn` is crucial to some of our attacks. The transformation `MixColumn` operates on the columns of a state separately. To each column a fixed linear transformation is applied. To do so, bytes are interpreted as elements in the field $\mathbb{F}_{2^8}$. As is usually done, we will denote elements in this field in hexadecimal notation. Hence $01, 02$ and 03 correspond to the bytes $00000001, 00000010$, and 00000011, respectively. Now `MixColumn` applies to each row of a state the linear transformation defined by the following matrix

$$\begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix}. \tag{1}$$

The operation `xtime`. The multiplications in $\mathbb{F}_{2^8}$ necessary to compute the transformation `MixColumn` are of great importance to some of our attacks. Therefore we are going to describe them in more detail. First we need to say a few words about the representation of the field $\mathbb{F}_{2^8}$. In AES the field $\mathbb{F}_{2^8}$ is represented as

$$\mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1). \tag{2}$$

That is, elements of $\mathbb{F}_{2^8}$ are polynomials over $\mathbb{F}_2$ of degree at most 7. The addition and multiplication of two polynomials is done modulo the polynomial $x^8 + x^4 + x^3 + x + 1$. Since this is an irreducible polynomial over $\mathbb{F}_2$, (2) defines a field. In this representation of $\mathbb{F}_{2^8}$ the byte $\mathbf{a} = (a_7, \ldots, a_1, a_0)$ corresponds to the polynomial $a_7x^7 + \cdots a_1x + a_0$. The multiplication of an element $\mathbf{a} = (a_7, \ldots, a_1, a_0)$ in $\mathbb{F}_{2^8}$ by $01, 02$, and 03 is realized by multiplying the polynomial $a_7x^7 + \cdots a_1x + a_0$ with the polynomials $1, x, x + 1$, respectively, and reducing the result modulo $x^8 + x^4 + x^3 + x + 1$. Hence

$$\begin{aligned} 01 \cdot \mathbf{a} &= \mathbf{a} \\ 03 \cdot \mathbf{a} &= 02 \cdot \mathbf{a} + \mathbf{a}. \end{aligned}$$

We see that multiplying a column of a state by the matrix in (1) can be reduced to a number of additions modulo 2 and multiplications by 02. We consider the multiplication by 02 in more detail. Following the notation in [DR2] we denote the multiplication of byte $\mathbf{a}$ by 02 by $\mathtt{xtime}(\mathbf{a})$. The crucial observation is that $\mathtt{xtime}(\mathbf{a})$ is simply a shift of byte $\mathbf{a}$, followed in some cases by an xor of two bytes. More precisely, for $\mathbf{a} = (a_7, \ldots, a_0)$

$$\mathtt{xtime}(\mathbf{a}) = \begin{cases} (a_6, \ldots, a_0, 0) & \text{if } a_7 = 0 \\ (a_6, \ldots, a_0, 0) \oplus (0, 0, 0, 1, 1, 0, 1, 1) & \text{if } a_7 = 1 \end{cases} \tag{3}$$

This finishes our brief description of the AES encryption procedure.

3 Physical Faults Attacks

Although there are lots of possibilities to introduce an error during the cryptographic operation of an unprotected smartcard IC, we will only briefly explain so called spike attacks, glitch attacks and the recently developed optical/eddy current attacks. We selected these attacks, as they are so called non invasive attacks or at least only semi-invasive, meaning that these methods require no physical opening, no chemical preparation nor an electrical contact to the chip's metal surface. Therefore, these attacks are the most obvious methods for attacking smartcard ICs by fault attacks. For a thorough treatment of tamper-resistance and especially on methods how to enforce erroneous computations of microcontroller chips we refer to [A,ABFHS,AK1,AK2,Gu1,Gu2,Koca,QS,SA,KK,Ma].

Moreover, although the effects of applying physical stress via the above mentioned methods to a smartcard IC can be completely explained by numerous physical and technical considerations, these explanations are clearly beyond the scope of this paper, meaning that our physical explanations will be pretty rough.

3.1 Spike Attacks

As required by [ISO], a smartcard IC must be able to tolerate on the contact V_{CC} a supply voltage between $4,5$V and $5,5$V, where the standard voltage is specified at 5V. Within this range the smartcard must be able to work properly. However, a deviation of the external power supply, called *spike*, of much more than the specified 10% tolerance might cause problems for a proper functionality of the smartcard IC. Indeed, it will most probably lead to a wrong computation result, provided that the smartcard IC is still able to finish its computation completely.

Although a spike seems from the above explanation very simple, a specific type of a power spike is determined by altogether nine different parameters. These nine parameters are determined by a combination of time, voltage values, and the shape of the voltage transition. This indicates the range of different parameters which an attacker can try to modify in order to induce a fault.

As spike attacks are non-invasive attacks, they are the most obvious method for inducing computational faults on smartcard ICs. In particular, they require no physical opening, no chemical preparation of the smartcard IC and do not require making an electrical contact to the IC's metal surface.

3.2 Glitch Attacks

Similar to above, [ISO] prescribes that a smartcard IC must be able to tolerate on the clock contact CLK a voltage for V_{IH} of $0,7V_{CC},\ldots,V_{CC}$ and for V_{IL} of $0V_{CC},\ldots,0,5V_{CC}$. Moreover, it must be also able to tolerate clock rise and clock fall times of 9% from the period cycle. Within this range the smartcard must be able to work properly. However, a deviation of the external CLK, most often called *glitch*, of much more than the specified tolerances clearly will cause problems for a proper functionality of the smartcard IC. Again, similar to voltage spikes there is huge range of different parameters determining a glitch, which can

be altered by an attacker to induce a faulty computation on a device performing a encryption.

Interestingly, a finely tuned clock glitch is able to completely change a CPU's execution behavior including the omitting of instructions during the executions of programs. For physical explanations of this nice effect we refer the interested reader to [AK1,AK2,KK]. According to [KK], around 1999 clock glitch attacks have been the simplest and most practical attacks.

3.3 Optical Attacks

Without doubt, light attacks now belong to the classical non invasive methods to realize faulty computations on smartcards by manipulating their non-volatile memory, i.e., their EEPROM, cf. [A,AK1,AK2,BS97,KK,Koca,NR,Ma,Pe]. Unfortunately, such light attacks could be used so far only in an unsystematic way, for e.g. flipping randomly selected bits from 1 to 0 with some non zero probability, cf. [BS97,Pai].

However, recently it was demonstrated by [SA] that by using focused camera flash-light on the right places of a microcontroller it is indeed possible to set or reset any individual bit of its memory at a time specified by the attacker. Such an attack enables a very timing-precise controlled transient fault on a given individual bit. Moreover, it is claimed by [SA] to be practical and seems to require only very cheap and simple public available photo equipment. Therefore, [SA] called it optical attack instead of light attack, as it mainly relies on using optical equipment.

3.4 Electromagnetic Perturbations Attacks

Originally, [SQ] demonstrated the use of a passive coil to measure the electromagnetic emission of a processor according to the general theory of side-channel attacks [CKN]. However, recently it was demonstrated by [QS] that by inducing a so called *eddy current* via an alternating current in a coil near a conducting surface on the right place of a processor or memory it is indeed possible to set or reset any individual bit at a time specified by the attacker. Thus, the use of an active coil enables another very timing-precise controlled transient fault on a given individual bit. Moreover, it is claimed by [QS] to be practical and seems to require only very cheap and simple public available equipment. Additionally, it has the interesting advantage over an optical attack that it does not require any opening of the chip. Indeed, it is possible to induce an eddy current through the plastic of a smartcard.

3.5 Differentiating the Physical Attacks

In order to sort the zoo of possible effects by physically stressing a chip, we will now characterize the above attacks concerning:

- Control on fault location.
- Precision of timing.

– Fault model.
– Number of faulty bits induced.

Control on fault location. For the issue of the control on the resulting fault location, one can clearly define three natural classes: no control, loose control and complete control. The definition of them is self contained.

Precision of timing. As with the former issue it is obvious to define three natural classes whose definition speaks for themselves: no control on the fault occurence time, loose control on the fault occurence time and very precise control on the fault occurence time.

Fault model. Although there are usually only three fault models specified, cf. [BDL,BDHJNT,BMM,BS97,YJ,YKLM1,YKLM2,ZM], we will introduce a new fault model. This model reflects the power of the optical/eddy current induced fault models as developed in [SA,QS]. In addition to the classical fault models, which are given by the stuck at faults model (saf), bit flip model (bf) or random fault model (rf), we will define the *bit set or reset model* (bsr). In this model the attacker is able to set or reset the value of any target bit at any time specified by himself, regardless of its previous contents, cf. [SA,QS].

Number of faulty bits induced. Clearly, the number of induced faulty bits due to physical stress is very important for a fault based cryptanalysis. Here one usually makes the distinction between single faulty bit, few faulty bits and random number of faulty bits.

We present the characterization of the different physical attacks presented previously by a table.

Table 1.

	Control on fault loc.	Prec. of timing	Fault model	#(faulty bits induc.)
Spike	no	precise	random	random
Glitch	depending	precise	depending	depending
Optical	complete	precise	bsr and bf	as required
Eddy current	complete	precise	saf, bsr and bf	as required

4 A General Fault Attack on AES

In this section we describe an implementation independent fault attack on AES. The attack is quite general and can be mounted against any symmetric encryption cipher that applies an initial key addition, also called key whitening. Ciphers using key whitening include IDEA, SAFER, and Blowfish.

First we will show how to compute the complete cipher key under the assumption that $\texttt{Nk} \leq \texttt{Nb}$, that is, the block length is greater or equal than the key size. In this case the complete cipher key is used in the initial `AddRoundKey`. We will show how to compute the cipher key bit by bit.

As described in Section 2.1 the cipher key is stored in a $4 \times \texttt{Nr}$ array of bytes (k_{ij}). We denote the l-th bit in byte k_{ij} by $k_{ij}^l, 0 \leq l < 7$. Similarly, the l-th bit in state byte a_{ij} will be denoted by a_{ij}^l. We will show how an attacker can determine k_{ij}^l by inducing a single fault during one encryption. Consider the plaintext $\mathbf{0} = 0^{8 \cdot \texttt{1b}}$, i.e., each bit of $\mathbf{0}$ has value 0. The attacker encrypts $\mathbf{0}$. During the initial transformation `AddRoundKey` the operation

$$a_{ij} := 0^8 \oplus k_{ij}$$

will be performed. Of course, after this operation has been performed, we have $a_{ij} = k_{ij}$. Before the next transformation is performed, the attacker tries to set a_{ij}^l to 0. After this fault has been induced, the encryption proceeds without further faults being induced.

The main observation is that if $k_{ij}^l = 0$, then setting the value of a_{ij}^l to 0, did not change the value of this bit and we still get a correct encryption of $\mathbf{0} = 0^{8 \cdot \texttt{1b}}$. However, if $k_{ij}^l = 1$, then setting the value of a_{ij}^l to 0 results in an incorrect ciphertext and the cryptographic device will answer with `reset`. Hence, from the behavior of the device the attacker can deduce the value of k_{ij}^l.

Altogether, we see that in case $\texttt{Nk} \leq \texttt{Nb}$ we can determine the complete cipher key by encrypting $8\texttt{lk}$ times the message $\mathbf{0}$, each time inducing a single fault.

Next we consider the case $\texttt{Nk} > \texttt{Nb}$. In case $\texttt{Nk} > \texttt{Nb}$, an attacker can determine the first `Nb` bytes of the cipher key using the attack described above. Once the attacker knows the first `Nb` blocks of the cipher key, he can simulate the AES encryption process up to the `AddRoundKey` transformation in round 1. More importantly, the attacker can compute a plaintext P, such that during the encryption of P the state (a_{ij}) prior to the application of `AddRoundKey` in round 1 consists of zero bytes only, i.e., $a_{ij} = 0^8$ for all i, j.

To determine a_{ij}^l for $j \geq \texttt{Nk}$, the attacker encrypts the plaintext P. After the transformation `AddRoundKey` has been performed in the first round, the attacker resets a_{ij}^l to 0. After that, the encryption procedure proceeds without further faults. As before and by choice of P, if $k_{ij}^l = 0$ resetting the value of a_{ij}^l has no effect and the correct encryption of P is computed. However, if $k_{ij}^l = 1$, the ciphertext will not be correct. As before, the attacker can determine the value of k_{ij}^l.

Recall that we always have $\texttt{Nk} \leq 2\texttt{Nb}$ (see Section 2.1). Hence in the way just described the attacker can determine the complete cipher key. Overall, to compute the cipher key, an attacker needs to compute the encryption of $8\texttt{lk}$ plaintexts, each time inducing a single fault.

4.1 A Probabilistic Fault Model

Inspired by technological considerations we will now consider a slightly relaxed fault model. Namely, given the rapid shrink processes of semiconductor technologies, it is plausible to assume that optical/eddy current attacks using a cheap and unprofessional lithography/coil equipment will not achieve a sufficient resolution to exactly attack single bits in the desired way. In the best case an optical/eddy current attack will approximately hit the correct physical location to set or reset a specific bit. We model this in the following way. If the value of the bit an attacker tries to reset is 1, then we assume that with probability $p_1 > \frac{1}{2}$ the attacker successfully resets the value of that bit to 0. If the value of the bit the attacker tries to reset is 0, then we assume that the value will change to 1 with probability $p_0 < \frac{1}{2}$. We can assume that the probabilities p_0, p_1 are known to the attacker.

We only describe how to determine the value of k_{00}^0 with this fault model. The generalization to arbitrary bits of the cipher key is straightforward. To compute k_{00}^0 the attacker encrypts the message $\mathbf{0}$ m-times, where m is a parameter to be specified later. In each encryption of $\mathbf{0}$, after the operation

$$a_{00} = 0^8 \oplus k_{00}$$

has been performed in the initial `AddRoundKey`, the attacker tries to reset the value of a_{00}^0. For each encryption the attacker can deduce from the behavior of the cryptographic device whether it yielded a correct or an incorrect ciphertext. If the number of incorrect ciphertexts is at least

$$m\left(\frac{p_1 + p_0}{2}\right)$$

the attacker guesses $k_{00}^0 = 1$. Otherwise, the attacker guesses $k_{00}^0 = 1$.

Let us analyze the probability that the attacker guesses k_{00}^0 correctly. If $k_{00}^0 = 1$, then in each encryption of $\mathbf{0}$ the value of a_{00}^0 will be set to 0 with probability p_1. Hence in the case $k_{00}^0 = 1$, each encryption of $\mathbf{0}$ results in an incorrect ciphertext with probability p_1. Therefore, we expect $p_1 m$ incorrect ciphertexts. By Bernstein's inequality (see for example [GS]), the probability that in this case fewer than $m\left(\frac{p_1+p_0}{2}\right)$ ciphertexts are incorrect is bounded from above by $\exp(-m(p_1 - p_0)^2/16)$.

Similarly, one can show that in case $k_{00}^0 = 0$, the probability that the number of incorrect ciphertexts is larger than $m\frac{p_1+p_0}{2}$ is also bounded from above by $\exp(-m(p_1 - p_0)^2/16)$. Altogether we obtain that the attacker guesses bit k_{00}^0 correctly with probability at least

$$1 - \exp\left(-m\frac{(p_1 - p_0)^2}{16}\right).$$

One checks that with the choice

$$m = \frac{176}{(p_1 - p_0)^2}$$

the attacker correctly guesses k_{00}^0 with probability $1 - 2^{-15}$. Analogously, the other cipher key bits can be determined. If for every cipher key bit we choose $m = 176/(p_1 - p_0)^2$ plaintexts, the probability that every cipher key bit key is guessed correctly is $1 - 2^{-8}$.

For example, if $p_1 = \frac{3}{4}$, $p_0 = \frac{1}{4}$ and $\mathtt{lk} = 16$, then with the choice $m = 90112$ the attacker guesses the complete cipher key correctly with probability $1 - 2^{-8}$.

We note that the bounds on the probability that the attacker guesses a bit incorrectly can be reduced somewhat by using Chernoff bounds instead of Bernstein's bound (see [GS]).

4.2 Relaxing the Timing Constraint

In the basic fault model as well as in the probabilistic fault model, we always assumed that the attacker is able to exactly determine the time when he resets the value of a memory bit b. Unfortunately, as described in [CCD], some modern secure microcontrollers are equipped with a randomized timing behavior to counteract statistical side-channel attacks. However, in this section we will show that under some very plausible assumptions this hardware mechanism doesn't yield sufficient protection. More specifically, we only want to assume that the attacker has a certain window of time in which he can reset the value of a particular memory bit b. We assume, that within the time window there are only c operations performed by the AES encryption procedure that involve bit b. The attacker knows c. Furthermore we assume that the time at which bit b is reset is randomly distributed. That is, for each $t = 0, \ldots, c$ the probability that b is reset after the execution of the t-th operation involving b, but before the following operation involving b, is $\frac{1}{c}$. We describe the modifications that have to applied to the basic attack to take care of this weaker assumption. Similar modifications can be carried out for the attack in the probabilistic fault model.

We only describe how to determine the first cipher key bit k_{00}^0. Instead of encrypting the plaintext $\mathbf{0}$, the attacker encrypts several plaintexts $P_1, \ldots, P_m$. In all these plaintexts P_i the leftmost bit has value 0. The remaining bits of the plaintexts are chosen uniformly at random. As before the attacker tries to reset the value of a_{00}^0 after the initial `AddRoundKey` has been performed. Instead he only manages to reset a_{00}^0 within a time window that besides the initial operation $a_{00} := a_{00} \oplus k_{00}$ includes $c - 1$ other operations involving a_{00}^0. However, the initial `AddRoundKey` is the first transformation involving a_{00}^0. The next operation involving a_{00}^0 is the `ByteSub` of round 1. We now make the following heuristic assumption:

Assume that bit a_{00}^0 has a fixed value b, while the remaining bits of a_{00} are chosen uniformly at random. Then the leftmost bit of `ByteSub`(a_{00}) *is distributed uniformly at random.*

In fact, a random function on 8 bits clearly has this property. Now, the transformation `ByteSub` is usually assumed to behave like a random function. This justifies the assumption.

From the assumption it follows that unless the attacker manages to reset a^0_{00} immediately after the initial `AddRoundKey` has been performed, the attacker tries to reset a bit whose value is distributed uniformly at random.

Next we compute the probability that the fault induced by the attacker during the encryption of plaintext P_i leads to an incorrect cipher text. With probability $\frac{1}{c}$ the attacker resets a^0_{00} immediately after the initial `AddRoundKey`. In this case, the ciphertext will not be correct if and only if $k^0_{00} = 1$. With probability $1 - \frac{1}{c} = \frac{c-1}{c}$ the attacker resets bit a^0_{00} following the `ByteSub` of round 1 or later. From the assumption stated above, it follows that in this case the attacker resets the value of a bit whose value is distributed uniformly at random. Therefore the ensuing ciphertext will not be correct with probability $\frac{1}{2}$.

We conclude that for all plaintexts P_i, if $k^0_{00} = 0$ the encryption of P_i will be incorrect with probability

$$\frac{c-1}{2c}.$$

On the other hand, if $k^0_{00} = 1$ then the ciphertext for P_i will not be correct with probability

$$\frac{c-1}{2c} + \frac{1}{c}.$$

As in the probabilistic fault model, this difference in probability can be exploited to guess the value of k^0_{00} correctly with high probability. The attacker simply chooses m large enough and depending on the number of incorrect ciphertexts he sets k^0_{00} to 0 or 1. The details are exactly as in the previous section, so we omit them.

5 Implementation Specific Fault Attacks

Since a fault-based cryptanalysis (actually an engineering type of attack) might also exploit some peculiar implementation properties, we now take a closer look at some particular implementation details to understand the fault-based vulnerability of AES. We will first present the idea for our implementation specific fault attacks. Then we we will apply this idea to several conceivable implementations of the `xtime` operation. Nevertheless, it should be clear that also other implementations of `xtime` might be suspectible to the kind of attack described below. Moreover, we would like to stress that the ideas developed within Sections 4.1 and 4.2 also apply to the following implementation specific fault attack scenarios.

5.1 Description of the Underlying Idea

Combining an idea of [YJ] together with ideas from the Timing Analysis of the AES, we will turn the attack of [KQ] into a fault based cryptanalysis of the AES. Depending on the actual realization of the `xtime` operation, different attack scenarios will follow.

Table based fault attacks of the AES. First we describe how to determine the first byte of the cipher key, when given the information that a specific `xtime` operation reduces its result by xoring it with the byte $(0,0,0,1,1,0,1,1)$ (see (3)). But, in contrast to [KQ] we will get this information by inducing a computational error during that specific `xtime` operation. The information retrieval process itself is the implementation dependent part of the attack. For some conceivable implementations it will be described later.

First a $256 \times N$ table T is set up. Here $N \leq 256$ is some parameter to be specified later. Every row in this table corresponds to a possible value for the first cipher key byte k_{00}, while the columns correspond to possible values for the first plaintext byte a_{00}. The table entries $T[k,m], 0 \leq k < 256, 0 \leq m < N$ are defined as follows

$$T[k,m] = \begin{cases} 1 \text{ if the leftmost bit of } \texttt{ByteSub}(k \oplus m) \text{ is } 1 \\ \\ 0 \text{ if the leftmost bit of } \texttt{ByteSub}(k \oplus m) \text{ is } 0 \end{cases} \tag{4}$$

Next the attacker constructs N plaintexts $m_0, \ldots, m_{N-1}$. The first byte of the plaintext m_i is i. The remaining text bytes are chosen uniformly at random. Now an attacker encrypts the messages m_i and for each i he enforces an error during the encryption of the plaintext. As already said, the actual time and kind of error will be specified below depending on the specific implementation of the `xtime` operation. At this point it suffices to note that during the encryption of plaintext m_i the transformation `MixColumn` in round 1 multiplies the byte $\texttt{ByteSub}(k_{00} \oplus i)$ by 02, or equivalently `MixColumn` applies $\texttt{xtime}(\texttt{ByteSub}(k_{00} \oplus i))$. This operation will be shown to be highly vulnerable to a fault attack revealing whether the leftmost bit of $\texttt{ByteSub}(k_{00} \oplus i)$ is 1 or 0. Hence, the attacker is then able to predict the leftmost bit of $\texttt{ByteSub}(k_{00} \oplus i)$.

Then the attacker compares his predictions for the leftmost bits of $\texttt{ByteSub}(k_{00} \oplus i), i = 0, \ldots, N-1$, with the entries in table T. If the attacker chooses $N = 256$ and his predecitions are correct, this comparison reveals the first cipher key byte. However, we expect that fewer than 256 plaintexts P_i will suffice to determine the first cipher key byte. In fact, Koeune and Quisquater observed that in their timing attack $N = 20$ already suffices. Since a fault attack will yield much more reliable predictions for the leftmost bits of $\texttt{ByteSub}(k_{00} \oplus i)$ we expect that in our case $N \approx 16$ will suffice.

Note that the main diagonal of the matrix in (1) consists of 02 only. Hence during `MixColumn` every state byte gets multiplied by 02 once. From this, one concludes that the method to compute the cipher key byte k_{00}^0 can be used to compute the other cipher key bytes as well. Due to the AES's key schedule as described in section 2.1, the cipher is broken once the attacker has determined `Nk` consecutive bytes of the round key. Thus, for $\texttt{Nk} \leq \texttt{Nb}$ the above methods breaks AES. For smaller block sizes, similiar ideas as presented in section 4 apply.

Based on this table approach we now present in figure 1 the resulting common skeleton for all of our following implementation specific fault attacks.

build table $T[k, m]$
build plaintexts $m_0, \ldots, m_N$

for $i := 0$ **to** N **do**
  encrypt the message m_i and
  disturb the operation `xtime`(`ByteSub`($k_{00} \oplus i$)) within `MixColumn`
  of round 1 by an appropriate physical attack as described later
  if output refused or incorrect **then**
    $\tilde{T}[i]$ is set to 0 or 1 depending on the `xtime` realization
od

by comparing the tables T and $\tilde{T}$ determine the first key byte k_{00}

output: k_{00}

Fig. 1. Common fault attack skeleton, revealing the first key byte.

5.2 The Simplest Attack against an Unskilled Textbook-Secured Implementation

The simplest and also most obvious way for a Timing Analysis (and SPA) resistant implementation of the `xtime` operation seems to be given by the following. Save the most significant bit of the input byte, shift the byte by one bit to the left, perform *two* xor's with the shifted byte. Finally, according to the most significant bit of the input byte, return one of the two previously computed results as shown in the figure 2.

input: $a = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$

$f := a_7$
$a := (a_6, a_5, a_4, a_3, a_2, a_1, 0)$
`xtime`$[0] := a \oplus (0, 0, 0, 0, 0, 0, 0, 0)$
`xtime`$[1] := a \oplus (0, 0, 0, 1, 1, 0, 1, 1)$
return(`xtime`$[f]$)

output: `xtime`(a)

Fig. 2. Timing Analysis secured `xtime`(`a`) realization.

Let us now analyze what happens, if an attacker uses the fault attack skeleton outlined above. That is, the attacker encrypts N plaintexts m_i, $i = 1, \ldots, N$. Moreover, in each encryption the attacker disturbs the computation of `xtime`$[0]$. Here we can apply a very liberal fault model. We assume that the attacker is able to enforce an arbitrary wrong value

$$\texttt{xtime}[0]' \neq \texttt{xtime}[0].$$

Then, depending on bit a_7 of the original byte a, the following will result. If $a_7 = 1$, due to the fact that `xtime`[1] will be returned, the wrong result `xtime`$[0]'$ is of no further interest and will be therefore discarded. Therefore the encryption yields a correct ciphertext. However, if $a_7 = 0$, the wrong result `xtime`$[0]'$ will lead to a wrong ciphertext and the cryptographic device will answer with `reset`. Hence, from the behavior of the device, the attacker can determine the bit a_7. Now the attacker can proceed as described in Section 5.1 to completely break the cipher.

Note, that we did not make any assumption on the kind of introduced error during the computation of `xtime`[0]. We simply need a wrong result. Thus it is indeed easy to realize by any one of the methods presented in 3, indicating that this attack is actually devestating.

5.3 An Attack Against a Possible Real Implementation

After we have seen an unskilled textbook implementation of the `xtime` operation, we will now consider a slow but real implementation written in an extended 8051 assembler language. We selected an extended 8051 microcontroller as it is the most commonly used controller in todays' smartcards. Consider the following `xtime` realization, which is obviously secure against a Timing Analysis.

```
; xtime(a)

; parameters: "a" in accu

; return value: "xtime(a)" in accu

xtime:

  MOV B, A                          ; B:=a
  SLL B                             ; B:=(a_6,a_5,...,a_1,0)
  DIV A, #10000000b                 ; A:=(0,0,...,0,a_7)
  MUL A, #00011011b                 ; A:=a_7*(0,0,0,1,1,0,1,1)
  XRL A, B                          ; A:=A⊕(a_6,a_5,...,a_1,0)
  RET
```

Fig. 3. Timing Analysis secured `xtime`(**a**) realization.

As above, let us analyze what happens, if an attacker applies the fault attack skeleton, i.e., he encrypts N plaintexts m_i, $i = 1, \ldots, N$. However, this time the attacker will apply a glitch attack, cf. Section 3.2, to omit the execution of the `MUL A, #00011011b` instruction.

Then, depending on bit a_7 of the input byte a, the following will result. If $a_7 = 0$, the register `A` will be zero after the instruction `DIV A, #10000000b`. Thus, omitting via a glitch the following `MUL A, #00011011b` instruction does

not matter, as it would write back to register `A` a zero, still guaranteeing a correct encryption of m_i. Hence, in case $a_7 = 0$, the encryption processes ends with a correct ciphertext. However, in the case of $a_7 = 1$, omitting the `MUL A, #00011011b` instruction, will result in a value 1 in register `A`, whereas $(0, 0, 0, 1, 1, 0, 1, 1)$ would be the correct value. Thus, the cryptographic device will detect a computational error and will answer with `reset`, resulting in an answer which indicates that a wrong computation happened which will actually be observed by the attacker. Therefore, from the behavior of the cryptographic device, the attacker can deduce the value of bit a_7. As before, applying in this fashion the fault attack skeleton to every one of the 32 key bytes will completely break the cipher.

5.4 An Attack against a Suggested Implementation

Now, that we have seen some vulnerable `xtime` realizations we will consider the implementation actually suggested by the inventors of the AES, as proposed in [DR2]. Inspired by the knowledge about the timing analysis vulnerability of the AES, they proposed to implement `xtime` as an array T consisting of 256 bytes, where

$$T[\mathbf{b}] := \texttt{xtime}(\mathbf{b}).$$

As above, an attacker applies the fault attack skeleton as described in Section 5.1. This time the attacker will apply an optical/eddy current attack, cf. Sections 3.3 and 3.4. Clearly it is most conceivable that the table T will be stored in ROM and therefore is of no use to an attacker. However, the whole *current* state of the AES encryption clearly must be stored in RAM. Therefore, this time the attacker will reset via an optical/eddy current attack on the RAM the bit a_7 of the state byte a.

Although the analysis is very similiar to the former two cases, for completeness sake we will include it. Depending on the bit a_7 of the input byte a, the optical/eddy current attack will have the following effect. If $a_7 = 0$, trying to reset it via optical/eddy current attacks has no effect. Therefore, the table T is consulted for the correct value and the cryptographic device will answer with a correct ciphertext. However, in case $a_7 = 1$, resetting this bit to 0 must result in a wrong encryption, since the table T is consulted for a wrong value. Hence, from the behavior of the cryptographic device an attacker can determine the value of bit a_7. As before described in Section 5.1 this can be used to completely break the cipher.

Although this attack looks very similar to the attacks described within section 4, it has definitely an advantage when compared with these attacks. Namely, here the attacker has to focus only on one single bit, i.e. a_7, whereas for the attacks described in section 4 he would have to move his equipment over all cipher bits. From a realization point of view, it seems much easier to concentrate a physical attack equipment precisely on one single location rather than on many ones.

Finally we would like to note that recently the table based approach for `xtime` was also shown to be susceptible to a Differential Power Analysis by [YJ]. Given all this, one can definitely say that the suggestion of [DR2] is completely insecure against physical side-channel attacks.

5.5 An Attack against a Hardware Implementation

So far we have only considered software realizations of `xtime`, eventually we will briefly strive over a possible hardware realization. However, as a complete hardware description even if only for the `MixColumn` transformation is clearly out of the present papers scope, we will concentrate again on the `xtime` circuit. In this vein of building a dedicated AES circuit, it is widely anticipated, cf. [Wo], that `xtime` should be realized by the following very simple circuit, being beside clearly secure against a Timing Analysis.

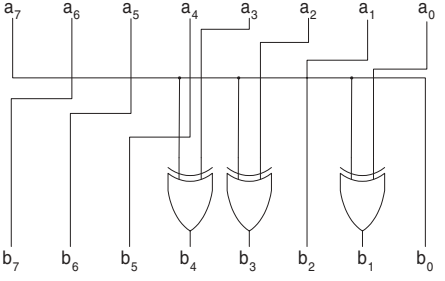

Fig. 4. Circuit for $\mathbf{b} := \texttt{xtime}(\mathbf{a})$.

We will now elaborate this situation a bit more in depth. Forced by the tough area requirements for a chipcard IC, most AES hardware architecture proposals are silicon size optimized. This means in particular, that such AES hardware modules really compute the AES operations `ByteSub`, `MixColumn` and of course `xtime`, instead of using large ROM based lookup tables to compute the corresponding transformations, cf. [SMTM,Wo,WOL]. However, due to the logical depth of the corresponding computations, those architectures have to take into account that every single transformation needs at least one clock cycle, and most often it will require more cycles. Thus, between different transformations the whole AES encryption state is stored within a register bank, which is actually realized by a large number of flip-flops, cf. [WE]. In particular, the whole encryption state prior to the execution of `MixColumn` must be stored within flip-flops, cf. [WE]. Indeed, the inputs $(a_7, \ldots, a_0)$ to a corresponding `xtime` circuit are actually flip-flops, storing the results from the previous `ShiftRow` transformation. But this in turn means that again we can use an optical/eddy current attack, cf. Sections 3.3 and 3.4, against these flip-flops, exactly as described in [SA]. Indeed, this time the attacker will reset (during the time period when the results from the previous `ShiftRow` are stored within the flip-flops) the bit a_7 of the state byte a via an optical/eddy current attack on the aforesaid flip-flop. The rest of the analysis is completely analog to the table based `xtime` realization.

6 Conclusions and Countermeasures

Counteracting fault attacks is usually done by some naive software countermeasures [BS97] or some proposed hardware architectures [KWMK]. However, both simply check the computed ciphertext for correctness and will cause the chip to react with some kind of alarm if an erroneous ciphertext has been detected. Unfortunately, as shown in the present paper, this is obviously detectable and exploitable to mount our fault attacks.

So far we are not aware of any software countermeasures for the AES which could provably prevent the attacks described in the present paper.

Nevertheless, we would like to point out that some modern high-end crypto smartcards are protected by various and numerous means of sophisticated hardware mechanisms to detect any intrusion attempt to their system behavior, cf. [Ma,MACMT,MAK,NR]. Various but not all hardware manufacturers of cryptographic devices such as smartcard ICs have been aware of the importance of protecting their chips against intrusions by, e.g., external voltage variations, external clock variations, light attacks, etc. To do so they use carefully developed logic families, cf. [MACMT,MAK], sensors, filters, regulators, etc. And indeed, only this special hardware countermeasures might give rise to a trustworthy functionality of the chip. The reason is that only those proprietary and secretly kept hardware mechanisms are indeed designed to counteract the source of a physical attack and not to counteract their effect on computations.

Acknowledgments. We would like to thank Alexander May for careful reading of our paper.

References

[A] R. Anderson, *Security Engineering*, John Wiley & Sons, New York, 2001.

[ABFHS] C. Aumüller, B. Bier, W. Fischer, P. Hofreiter, J.-P. Seifert, "Fault attacks on RSA: Concrete results and practical countermeasures", *Proc. of CHES '02*, Springer LNCS, pp. 261–276, 2002.

[AG] M. L. Akkar, C. Giraud, "An implementation of DES and AES, secure against some attacks", *Proc. of CHES '01*, Springer LNCS vol. 2162, pp. 315–324, 2001.

[AK1] R. Anderson, M. Kuhn, "Tamper Resistance – a cautionary note", *Proc. of 2nd USENIX Workshop on Electronic Commerce*, pp. 1–11, 1996.

[AK2] R. Anderson, M. Kuhn, "Low cost attacks attacks on tamper resistant devices", *Proc. of 1997 Security Protocols Workshop*, Springer LNCS vol. 1361, pp. 125–136, 1997.

[BDL] D. Boneh, R. A. DeMillo, R. Lipton, "On the Importance of Eliminating Errors in Cryptographic Computations" *Journal of Cryptology* **14**(2):101–120, 2001.

[BDHJNT] F. Bao, R. H. Deng, Y. Han, A. Jeng, A. D. Narasimbalu, T. Ngair, "Breaking public key cryptosystems on tamper resistant dives in the presence of transient faults", *Proc. of 1997 Security Protocols Workshop*, Springer LNCS vol. 1361, pp. 115–124, 1997.

[BS97] E. Biham, A. Shamir, "Differential fault analysis of secret key cryptosystems", *Proc. of CRYPTO '97*, Springer LNCS vol. 1294, pp. 513–525, 1997.

[BS99] E. Biham, A. Shamir, "Power analysis of the key scheduling of the AES candidates", *Proc. of the second AES conference*, pp. 115–121, 1999.

[BMM] I. Biehl, B. Meyer, V. Müller, "Differential fault attacks on elliptic curve cryptosystems", *Proc. of CRYPTO '00*, Springer LNCS vol. 1880, pp. 131–146, 2000.

[CCD] C. Clavier, J.-S. Coron, N. Dabbous, "Differential Power Analysis in the presence of Hardware Countermeasures", *Proc. of CHES '00*, Springer LNCS vol. 1965, pp. 252–263, 2000.

[CJRR] S. Chari, C. Jutla, J. R. Rao, P. J. Rohatgi, "A cautionary note regarding evaluation of AES candidates on smartcards", *Proc. of the second AES conference*, pp. 135–150, 1999.

[CKN] J.-S. Coron, P. Kocher D. Naccache, "Statistics and Secret Leakage", *Proc. of Financial Cryptography (FC '00)*, Springer LNCS, 2000.

[DR1] J. Daemen, V. Rijmen, "Resistance against implementation attacks: a comparative study", *Proc. of the second AES conference*, pp. 122–132, 1999.

[DR2] J. Daemen, V. Rijmen, *The Design of Rijndael*, Springer-Verlag, Berlin, 2002.

[GS] G. R. Grimmett, D. R. Stirzaker, *Probability and random processes*, Oxford Science Publications, Oxford, 1992.

[Gu1] P. Gutmann, "Secure deletion of data from magnetic and solid-state memory", *Proc. of 6th USENIX Security Symposium*, pp. 77–89, 1997.

[Gu2] P. Gutmann, "Data Remanence in Semiconductor Devices", *Proc. of 7th USENIX Security Symposium*, 1998.

[ISO] International Organization for Standardization, "ISO/IEC 7816-3: Electronic signals and transmission protocols", `http://www.iso.ch`, 2002.

[KR] B. Kaliski, M. J. B. Robshaw, "Comments on some new attacks on cryptographic devices", *RSA Laboratories Bulletin* **5**, July 1997.

[KK] O. Kömmerling, M. Kuhn, "Design Principles for Tamper-Resistant Smartcard Processors", *Proc. of the USENIX Workshop on Smartcard Technologies*, pp. 9–20, 1999.

[KQ] F. Koeune, J.-J. Quisquater, "A timing attack against Rijndael", *Université catholique de Louvain*, TR CG-1999/1, 6 pages, 1999.

[Koca] O. Kocar, "Hardwaresicherheit von Mikrochips in Chipkarten", *Datenschutz und Datensicherheit* **20**(7):421–424, 1996.

[KWMK] R. Karri, K. Wu, P. Mishra, Y. Kim, "Concurrent error detection of fault-based side-channel cryptanalysis of 128-bit symmetric block ciphers", *Proc. of IEEE Design Automation Conference*, pp. 579–585, 2001.

[Ma] D. P. Maher, "Fault induction attacks, tamper resistance, and hostile reverse engineering in perspective", *Proc. of Financial Cryptography*, Springer LNCS vol. 1318, pp. 109–121, 1997.

[Me] T. Messerges, "Securing the AES finalists against power analysis attacks", *Proc. of Fast Software Encryption 2000*, Springer LNCS vol. 1978, pp. 150–164, 2001.

[MAK] S. W. Moore, R. J. Anderson, M. G. Kuhn, "Improving Smartcard Security using Self-Timed Circuit Technology", *Fourth AciD-WG Workshop*, Grenoble, ISBN 2-913329-44-6, 2000.

[MACMT] S. W. Moore, R. J. Anderson, P. Cunningham, R. Mullins, G. Taylor, "Improving Smartcard Security using Self-Timed Circuit Technology", *Proc. of Asynch 2002*, IEEE Computer Society Press, 2002.

[NR] D. Naccache, D. M'Raihi, "Cryptographic smart cards", *IEEE Micro*, pp. 14–24, 1996.

[Pai] P. Pailler, "Evaluating differential fault analysis of unknown cryptosystems", *Gemplus Corporate Product R&D Division*, TR AP05-1998, 8 pages, 1999.

[Pe] I. Petersen, "Chinks in digital armor — Exploiting faults to break smartcard cryptosystems", *Science News* **151**(5):78–79, 1997.

[QS] J.-J. Quisquater, D. Samyde, "Eddy Current for Magnetic Analysis with Active Sensor", *Proc. of Int. Conf. on Research in SmartCards (E-Smart 2002)*, NOVAMEDIA, pp. 185–194, 2002.

[SQ] D. Samyde, J.-J. Quisquater, "ElectroMagnetic Analysis (EMA): Measures and Countermeasures for Smart Cards", *Proc. of Int. Conf. on Research in Smart Cards (E-Smart 2001)*, Springer LNCS vol. 2140, pp. 200–210, 2001.

[SMTM] A. Satoh, S. Morioka, K. Takano, S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization", *Proc. of ASIACRYPT '01*, Springer LNCS, pp. 241–256, 2001.

[SA] S. Skorobogatov, R. Anderson, "Optical Fault Induction Attacks", *Proc. of CHES '02*, Springer LNCS, pp. 2–12, 2002.

[WE] N. H. E. Weste, K. Eshraghian, *Principles of CMOS VLSI Design*, 2nd ed., Addison-Wesley, Reading MA, 1994.

[Wo] J. Wolkerstorfer, "An ASIC implementation of the AES MixColumn-operation", Graz University of Technology, Institute for Applied Information Processing and Communications, Manuscript, 4 pages, 2001.

[WOL] J. Wolkerstorfer, E. Oswald, M. Lamberger, "An ASIC implementation of the AES S-Boxes", *Proc. of CT-RSA Conference 2002*, Springer LNCS vol. 2271, 2002.

[YJ] S.-M. Yen, M. Joye, "Checking before output may not be enough against fault-based cryptanalysis", *IEEE Trans. on Computers* **49**:967–970, 2000.

[YKLM1] S.-M. Yen, S.-J. Kim, S.-G. Lim, S.-J. Moon, "RSA Speedup with Residue Number System immune from Hardware fault cryptanalysis", *Proc. of the ICISC 2001*, Springer LNCS, 2001.

[YKLM2] S.-M. Yen, S.-J. Kim, S.-G. Lim, S.-J. Moon, "A countermeasure against one physical cryptanalysis may benefit another attack", *Proc. of the ICISC 2001*, Springer LNCS, 2001.

[YT] S.-M. Yen, S. Y. Tseng, "Differential power cryptanalysis of a Rijndael implementation", LCIS Technical Report TR-2K1-9, Dept. of Computer Science and Information Engineering, National Central University, Taiwan, 2001.

[ZM] Y. Zheng, T. Matsumoto, "Breaking real-world implementations of cryptosystems by manipulating their random number generation", *Proc. of the 1997 Symposium on Cryptography and Information Security*, Springer LNCS, 1997.

Economics, Psychology, and Sociology of Security

Andrew Odlyzko

Digital Technology Center, University of Minnesota,
499 Walter Library, 117 Pleasant St. SE,
Minneapolis, MN 55455, USA
odlyzko@umn.edu
http://www.dtc.umn.edu/~odlyzko

Abstract. Security is not an isolated good, but just one component of a complicated economy. That imposes limitations on how effective it can be. The interactions of human society and human nature suggest that security will continue being applied as an afterthought. We will have to put up with the equivalent of bailing wire and chewing gum, and to live on the edge of intolerable frustration. However, that is not likely to to be a fatal impediment to the development and deployment of information technology. It will be most productive to think of security not as a way to provide ironclad protection, but the equivalent of speed bumps, decreasing the velocity and impact of electronic attacks to a level where other protection mechanisms can operate.

1 Introduction

This is an extended version of my remarks at the panel on "Economics of Security" at the Financial Cryptography 2003 Conference. It briefly outlines some of the seldom discussed-reasons security is and will continue to be hard to achieve.

Computer and communication security attract extensive press coverage, and are prominent in the minds of government and corporate decision makers. This is largely a result of the growing stream of actual attacks and discovered vulnerabilities, and of the increasing reliance of our society on its information and communication infrastructure. There are predictions and promises that soon the situation will change, and industry is going to deliver secure systems. Yet there is little visible change. Moreover, the same predictions and promises have been around for the last two decades, and they have not been fulfilled. At the same time, the world has not come to a grinding halt. Not only that, but, contrary to other predictions, there have not even been giant disasters, such as a failure of a bank, caused by information systems insecurity. The really massive financial disasters of the last few years, such as those at Enron, Long Term Capital Management, or WorldCom, owed nothing to inadequacy of information security systems. How can we explain this?

Growing ranks of observers have been arguing that one needs to understand the non-technical aspects of security, especially economic ones [1], [2], [4], [8],

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 182–189, 2003.
© Springer-Verlag Berlin Heidelberg 2003

[10]. Security does not come for free, and so it is necessary to look at the tradeoffs between costs and benefits. Furthermore, it is necessary to look at the incentives of various players, as many have an interest in passing on the costs of security to others, or of using security for purposes such as protecting monopolies. There is now even a series of workshops on Economics and Information Security (see [11] for information about the first one, including abstracts and complete papers). This note does not attempt to summarize the literature in this area. Instead, it briefly outlines some factors drawn from psychology and sociology that make security costly to implement, and thus require the economic tradeoffs that we observe being made. It also helps explain how we manage to live with insecure systems.

The basic problem of information security is that people and formal methods do not mix well. One can make the stronger claim that people and modern technology do not mix well in general. However, in many situations people do not have to be intimately involved with technology. If a combinatorial optimization expert finds a way to schedule airplanes to waste less time between flights, society will benefit from the greater efficiency that results. However, all that the passengers are likely to notice is that their fares are a bit lower, or that they can find more convenient connections. They do not have to know anything about the complicated algorithms that were used. Similarly, to drive over a bridge, all we need is an assurance that it is safe, and we do not require personal knowledge of the materials in the bridge. The fact that it took half as much steel to construct the bridge as it might have taken a century ago is irrelevant. We simply benefit from technology advances without having to be know much about them.

With security, unfortunately, technology can be isolated from people only up to a certain point. (The success of SSL/TLS was due to a large extent to its workings being hidden from users, so they did not have to do much to take advantage of it. That was an unusual situation, though.) As information technology permeates society, more and more people are involved. A system is only as secure as its weakest link, and in most cases people are the weak link. A widely circulated piece of Internet humor is the "Honor System Virus:"

> This virus works on the honor system.
>
> Please forward this message to everyone you know, then delete all the files on your hard disk.
>
> Thank you for your cooperation.

We can laugh at this, but in practice there are email messages popping up all the time, telling people that their computer has been infected by a virus, and telling them to find a file named "aol.exe" or something a bit more obscure, and to delete it. Moreover, a certain number of people do follow such instructions, and then plaintively call for help when they cannot connect to AOL. This is part of a pervasive phenomenon, in which people continue to lose money to the Nigerian 419 scam ("please help me transfer $36 million out of Liberia, and I will give you 20 percent"). Social engineering ("this is Joe from computer support,

we need your password to fix a bug in your computer") continues to be one of the most fruitful attack methods.

The standard response of technologists is to call for more and better education. However, that has not worked in the past, and is not likely to work in the future. Although education is useful, there will be countervailing tendencies (similar to those cited in [7]), namely more people will be using information and communication systems in the future, and those systems will be growing in complexity.

The message of this note is not that we should adopt a defeatist attitude to information security. The point is that we should be realistic about what can be accomplished. A productive comparison might be with auto safety. There has been substantial improvement in the past, and it is continuing. Greater crashworthiness of cars as well as better engineering of roads and more effective enforcement of drunk driving laws and more use of seat belts have made car travel far safer. In the United States, deaths per mile traveled by car fell at a compound annual rate of 4.7 percent between 1980 and 2000, by a cumulative factor of more than 2. However, because of growth in volume of travel (by 80 percent), the total number of deaths has only decreased from 50,000 per year to 42,000 per year. Moreover, in the last few years, the annual number of fatalities appears to have stabilized. Our society has decided (implicitly, without anyone ever voting on this explicitly) that we are willing to tolerate those 42 thousand deaths per year. Measures such as a drastic reduction in the speed limit, or devices that would constantly test the driver for sobriety or alertness, are not acceptable. Thus we manage to live with the limitation of the large masses of human drivers.

In information and communication technologies, we have also managed to live with insecurity. Chances are that we will manage to live quite well even with the projected insecurity of future systems. After all, we have lived without perfect security in the physical world. The locks on our house doors are not secure, nor are our electricity and water supply systems. In practice, though, existing safeguards are sufficient. Now the problem in cyberspace is that attacks can be mounted much faster and on a more massive scale than in the physical realm. The answer to that, though, is not to strive to build perfectly secure systems, as that is impossible. Instead, it should suffice to put in enough "speed bumps" to slow down attacks and keep their impact manageable. The reason this approach should work is the same one it has worked in the physical world, namely that it is not just the content of communication that matters, but also its context, and the economic, social, and psychological factors that hinder the deployment of secure systems provide protective mechanisms.

2 The Incompatibility of Formal Methods and Human Nature

A crippling problem for secure systems is that they would make it impossible for secretaries to forge their bosses' signatures. As was mentioned in [8], good secre-

taries know when it is safe to sign for their bosses to keep those bosses' workload manageable and speed the flow of work. There is some anectodal evidence that in organizations that move towards paperless offices, managers usually share their passwords with their secretaries, which destroys the presumed security of those systems. In a formal system, one can try to provide similar flexibility by building in delegation features. However, based on prior experience, it seems unlikely that one could achieve both security and acceptable flexibility.

In general, people like to have some slack in their lives. Sometimes this is exploited on purpose. Stuart Haber (private communication) reports that in marketing the digital time-stamping technology that he and Scott Stornetta invented, some accountants did raise concerns about losing the ability to backdate documents. As another example, when the U.S. Securities and Exchange Commission responded in 2002-2003 to the pressure to clean up corporate financial abuses, it attempted to make lawyers responsible for reporting malfeasance they encountered. The proposed wording of the rule had the definition [5]

> Evidence of a material violation means information that would lead an attorney reasonably to believe that a material violation has occurred, is occurring, or is about to occur.

However, lawyers objected to something this straightforward, and managed to replace it by

> Evidence of a material violation means credible evidence, based upon which it would be unreasonable, under the circumstances, for a prudent and competent attorney not to conclude that it is reasonably likely that a material violation has occurred, is ongoing, or is about to occur.

We can of course laugh at lawyers, but our lives are full of instances where we stretch the rules. Who does not feel aggrieved when they receive a speeding ticket for going 40 when the speed limit is 35?

There are also deeper sources of ambiguity in human lives. For example, suppose that you need to go to work, and you leave the key to your house or apartment with your neighbor, with the message "Please let in the plumber to fix the water heater." Seems like a very simple and well-defined request. However, suppose that after letting in the plumber, the neighbor sees the plumber letting in an electrician. You would surely expect your neighbor to accept this as a natural extension of your request. Suppose, though, that your neighbor then saw the plumber and the electrician carrying your furniture out. Surely you would expect the neighbor to call the police in such cases. Yet the request was simply "Please let in the plumber to fix the water heater." It did not say anything about calling the police. Human discourse is based on a shared culture, and the message "Please let in the plumber to fix the water heater" embodies expectations based on that culture. That is why we do not leave our keys with our neighbors' 6 year old daughter with such a message.

A particularly illustrative example of human problems with formal systems and formal reasoning is presented by the Wason selection task. It is one of

the cornerstones of evolutionary psychology. (See [3] for more information and references.) In this task, experimental subjects are shown four cards lying on a table. Each card is about a particular individual, Alice, Bob, Charlie, or Donna, and on on each side a statement about act by that individual. The subject's task is to decide, after reading the top sides of the cards, which of these cards need to be turned over to find out whether that individual satisfied some explicit rule. For example, we might be told that Alice, Bob, Charlie, and Donna all live in Philadelphia, and the rule might be that "If a person travels from Philadelphia to Chicago, he or she flies." For each person, one side of the card states where that person went, the other how they got there. The top side of Alice's card might say that she traveled to Baltimore, Bob's might say that he drove a car, Charlie's that he went to Chicago, and Donna's that she flew. For a logically minded person, it is clear that it is precisely Bob's and Charlie's cards that have to be turned over. In practice, though, only about a quarter of all subjects manage to figure this out.

The surprising part of the Wason selection task is what happens when the problem is restated. Suppose that now Alice, Bob, Charlie, and Donna are said to be children in a family, and the parents have a rule that "If a child has ice cream for dessert, he or she has to do the dishes after the meal." Suppose next that the top side of Alice's card states that she had fruit for dessert, Bob's that he watched TV after the meal, Charlie's that he had ice cream, and Donna's that she did the dishes. Most technologists immediately say that this is exactly the same problem as before, with only the wording changed. The rule is still of the form "If X then Y," only X and Y are different in the two cases, so again it is precisely Bob's and Charlie's cards that have to be turned over to check whether the rule is satisfied. Yet, among the general population, about three quarters manage to get this task right, in comparison to just one quarter for the earlier version. This (together with other experiments with other wordings and somewhat different settings) is interpreted as indicating that we have specialized mental circuits for detecting cheating in social settings.

The extended discussion of most people's difficulties with formal methods is motivated by the fact that security systems are conceived, developed, and deployed by technologists. They are among the small fraction of the human race that is comfortable with formal systems. They usually have little patience for human factors and social relations. In particular, they tend to expect others to think the way they do, and to be skilled at the formal thinking that the design and proper operation of secure systems require.

While people do have trouble with formal reasoning, we should not forget that they are extremely good at many tasks that computers are poor at. Just about any four year old girl is far superior in the ability to speak, understand spoken language, or recognize faces to even the most powerful and sophisticated computer system we have been able to build. Such abilities enable people to function in social settings, and in particular to cope with insecure systems. In particular, since information and communication systems do not operate in iso-

lation, and instead are at the service of a complicated society, there is a context to most electronic transactions that provides an extra margin of safety.

3 Digital Signatures versus Fax Signature

The 1980s were the golden age of civilian research on cryptography and security. The seeds planted in the 1970s were sprouting, and the technologists' bright hopes for a brave new world had not yet collided with the cold reality as clearly as they did in the 1990s. Yet the 1980s were also the age of the fax, which became ubiquitous. With the fax, we got fax signatures. While security researchers were developing public key infrastructures, and worrying about definitions of digital signatures, fax signatures became widespread, and are now playing a crucial role in the economy. Yet there is practically nothing as insecure as a fax signature, from a formal point of view. One can easily copy a signature from one document to another and this will be imperceptible on a fax. So what lessons can we draw from fax signatures, other than that convenience trumps security? One lesson is that the definition of a signature is, as with the message "Please let in the plumber to fix the water heater," loaded with cultural baggage that is hard to formalize.

It turns out that there is no strict legal definition of ordinary signature. We may think we know what a valid signature is, but the actual situation is quite complicated. An "X" may very well be a valid signature, even if it comes from somebody who normally signs her name in full. (She may have her hand in a cast, for example.) On the other hand, a very ordinary signature may not be valid, say if the signer was drunk while making it, or had a gun held to her head.

Furthermore, any signature, digital or physical, even if made willingly, may not be regarded as valid for legal enforcement of contract. Minors are not allowed to enter into most contracts. Even adults are not allowed to carry out some contracts that are regarded as against social policy, such as selling themselves or their children into slavery. Our legal system embodies many cultural norms (which vary from society to society, and even within a society change with time).

Another lesson is that our society somehow managed to function even when signatures became manifestly less secure with the spread of fax signatures. Moreover, it is easy to argue that fax signatures have contributed greatly to economic growth. How did this happen? This occurred because there is a context to almost every fax communication.

4 Social, Legal, and Economic Checks and Balances

Although fax signatures have become widespread, their usage is restricted. They are not used for final contracts of substantial value, such as home purchases. That means that the insecurity of fax communication is not easy to exploit for large gain. Additional protection against abuse of fax insecurity is provided by the context in which faxes are used. There are records of phone calls that carry the faxes, paper trails inside enterprises, and so on. Furthermore, unexpected large

financial transfers trigger scrutiny. As a result, successful frauds are not easy to carry out by purely technical means. Insiders (as at Enron and WorldCom and innumerable other enterprises) are much more dangerous.

Our commercial, government, and academic enterprises are large organizations with many formal rules and regulations. Yet the essential workings of these enterprises are typically based on various social relations and unwritten rules. As a result, one of the most effective tactics that employees have in pressuring management in labor disputes is to "work to rule." In general, the social and organizational aspects of large enterprises and even whole economies are poorly understood and underappreciated. Standard quantitative measures of invested capital or access to technology do not explain phenomena such as the continuing substantial lag of the regions of former East Germany behind former West Germany. There are other puzzling observations, such as the typical lack of measurable impact on economic output from major disruptions, such as earthquakes and snowstorms. There are ongoing attempts to understand just how societies function, including explorations of novel concepts such as "social capital." In general, though, it has to be said that our knowledge is still slight.

Information and communication technologies do play a crucial role in enabling smooth functioning of our complicated society, but are just a small part of it. That provides natural resilience in the face of formal system insecurities. Furthermore, the same limitations that make it hard to design, deploy, and effectively run secure systems also apply to attackers. Most criminals are stupid. Even those that are not stupid find it hard to observe the security precautions that are required for successful crime (such as inconspicuous consumption of their illicit gains). Even as determined an attacker as al Qaeda has had numerous security breaches. And, of course, the usual economic incentives apply to most attackers, namely that they are after material gains, have limited resources, and so on.

The natural resilience of human society suggests yet again the natural analogies between biological defense systems and technological ones. An immune system does not provide absolute protection in the face of constantly evolving adversaries, but it provides adequate defense most of the time.

The standard thinking in information security has been that absolute security is required. Yet we do have a rapidly growing collection of data that shows the value of even imperfect security. The experience of the pay-TV industry is certainly instructive. Although their systems have been cracked regularly, a combination of legal, technological, and business methods has kept the industry growing and profitable. Some more examples are offered by the applications of encryption technologies to provide lock-in for products, as in the replacement printer cartridge market [9]. Very often, "speed bumps" is all that is needed to realize economic value.

5 Conclusions

The general conclusion is that there is no "silver bullet" for security. In a society composed of people who are unsuited to formally secure systems, the best we

can hope to do is to provide "speed bumps" that will reduce the threat of cyberattacks to that we face from more traditional sources.

References

1. Anderson, R.J.: Liability and Computer Security – Nine Principles. ESORICS 94. Available at ⟨http://www.cl.cam.ac.uk/∼rja14⟩.
2. Anderson, R.J.: Security Engineering – A Guide to Building Dependable Distributed Systems. Wiley, 2001.
3. Cosmides, L., Tooby, J.: Evolutionary Psychology: A Primer. Available at ⟨www.psych.ucsb.edu/research/cep/primer.html⟩.
4. Geer, D.: Risk Management is Where the Money Is. Risks Digest, vol. 20, no. 6, Nov. 12, 1998. Available at ⟨http://catless.ncl.ac.uk/Risks/20.06.html⟩.
5. Norris, F.: No positives in this legal double negative. New York Times, January 24, 2003.
6. Odlyzko, A.M.: The Bumpy Road of Electronic Commerce. In: Maurer, H. (ed.): WebNet 96 – World Conf. Web Soc. Proc. AACE (1996) 378–389. Available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
7. Odlyzko, A.M.: The Visible Problems of the Invisible Computer: A Skeptical Look at Information Appliances. First Monday, **4** (no. 9) (Sept. 1999), ⟨http://www.firstmonday.org/issues/issue4_9/odlyzko/index.html⟩. Also available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
8. Odlyzko, A.M.: Cryptographic Abundance and Pervasive Computing. iMP: Information Impacts Magazine, June 2000, ⟨http://www.cisp.org/imp/june_2000/06_00odlyzko-insight.htm⟩. Also available at ⟨http://www.dtc.umn.edu/∼odlyzko/doc/recent.html⟩.
9. Static Control Corporation: Computer Chip Usage in Toner Cartridges and Impact on the Market: Past, Current and Future. White paper, dated Oct. 23, 2002, available at ⟨http://www.scc-inc.com/special/oemwarfare/default.htm⟩.
10. Schneier, B.: Secrets and Lies: Digital Security in a Networked World. Wiley, 2000.
11. Workshop on Economics and Information Security: May 16-17, 2002. Program and papers or abstracts available at ⟨http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/⟩.

Timed Fair Exchange of Standard Signatures
[Extended Abstract]

Juan A. Garay and Carl Pomerance

Bell Labs – Lucent Technologies, 600 Mountain Ave, Murray Hill, NJ 07974
{garay,carlp}@research.bell-labs.com

Abstract. In this paper we show how to achieve timed fair exchange of digital signatures of standard type. Timed fair exchange (in particular, contract signing) has been considered before, but only for Rabin and RSA signatures of a special kind. Our construction follows the gradual release paradigm, and works on a new "time" structure that we call a *mirrored time-line*. Using this structure, we design a protocol for the timed fair exchange by two parties of arbitrary values (values lying on their respective mirrored time-lines). We then apply the blinding techniques of Garay and Jakobsson to turn this protocol into a protocol for the timed fair exchange of standard signatures.
The length of these mirrored time-lines makes another problem apparent, which is making sure that the underlying sequence has a period large enough so that cycling is not observed. We also show how to construct these structures so that, under reasonable assumptions, this is indeed the case.

Keywords: Timed-release cryptography, timed commitments, contract signing, blind signatures.

1 Introduction

The exchange of digital signatures, and, in particular, contract signing, where the signatures are on a common piece of text, constitutes an important part of any business transaction, especially in settings such as the World Wide Web, where participants do not trust each other to some extent already. Recently, considerable efforts have been devoted to develop protocols that mimic the features of "paper contract signing," especially *fairness*. A contract signing protocol, or, more generally, an exchange of digital signatures, is *fair* if at the end of the protocol, either both parties have valid signatures, or neither does. In some sense, this corresponds to the "simultaneity" property of traditional paper contract signing. That is, a paper contract is generally signed by both parties at the same place and at the same time, and thus is fair.

Early work on fair exchange of secrets/signatures focused on the gradual release of secrets to obtain simultaneity, and thus fairness [Blu83,EGL85,Gol83] (see [Dam95] for more recent results). The basic idea is that if each party alternately releases a small portion of the secret, then neither party has a considerable

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 190–207, 2003.
© Springer-Verlag Berlin Heidelberg 2003

advantage over the other. Unfortunately, such a solution has several drawbacks in real situations. One problem is that of uncertain termination: if the protocol stops prematurely and one of the participants does not receive a message, he will never be sure whether the other party is continuing with the protocol, or has stopped—and perhaps even has engaged in another contract signing protocol! The other problem is how to enforce that neither party has a considerable advantage over the other. If the method is not designed properly, the party with more powerful computational resources could abort and employ all of his resources to complete the computation (e.g., recover the signature by parallel search of the remaining bits), while it might take much longer or even be infeasible for the weaker party to do so.

These problems have recently been tackled by Boneh and Naor [BN00] using tools based on *moderately-hard problems* [DN92]. (A moderately-hard problem is one which is not computationally infeasible to solve, but also not easy.) They propose an elegant "timing" mechanism based on modular exponentiation, which is a problem believed not to be well suited for parallelization. Indeed, considerable efforts have been invested in finding efficient exponentiation algorithms and still the best methods are sequential. Another important property of this mechanism is verifiability of the amount of work (number of steps) that would guarantee that a certain value is obtained. Using this mechanism, they introduce a variety of timed primitives, including timed commitments, timed signatures, and in particular, timed contract signing, where they show how to fairly exchange Rabin and RSA signatures of a special kind (namely, with modulus that is a Blum integer that fits the time structure).

In this paper we show how to achieve fair exchange—and contract signing—of standard signatures (e.g., RSA, DSA, Schnorr) without the modulus restriction above; more specifically, of signatures that allow for blinding [Cha82]. We build on the construction for timed release of standard signatures due to Garay and Jakobsson [GJ02]; however, the timed release is a "one way" operation, from a signer to a receiver, and cannot be directly applied to achieve a fair exchange. We answer this challenge by introducing a new time structure, which we call a *mirrored time-line*. In a nutshell, Garay and Jakobsson called a time-line the [BN00] structure (vector) of the form:

$$\{g^{2^{2^i}}\}_{i=0}^{k} \pmod{N}, \tag{1}$$

for N a Blum integer and generator g satisfying certain properties, and used an undisclosed value in the vicinity of the kth point (specifically, the kth point's square root) as the signature's blinding factor. A *mirrored time-line* is basically obtained by "concatenating" a time-line with its symmetric image (in other words, in the first half of a mirrored time-line the distance between the points increases exponentially, while in the second half it *decreases* like-wise). We then design a protocol—a "walk" on the mirrored time-line—which allows each party to successively approach the other party's undisclosed value in a synchronized

way. Finally, by using the blinding techniques of [GJ02], we obtain a protocol for the timed fair exchange of standard signatures.[1]

Increasing the length of these time structures poses another problem, which is making sure that the underlying sequences do not cycle, or at least that their period is large enough, otherwise no guarantees could be given that a time-line would be traversed sequentially. (In fact, this is also a problem with "regular," shorter time-lines, but our construction makes it more apparent.) We also show in this paper how to construct time-lines so that, under reasonable assumptions, the period of the underlying sequence will be large enough so that no cycling will occur. To our knowledge, although work has been done estimating the period of more general sequences [FPS01a,FPS01b], the case of the period of sequences such as the ones above has not been considered before.

Prior and related work. We already mentioned the work on fair exchange and contract signing based on the gradual exchange approach. An alternative approach to achieve fairness instead of relying on the gradual release of secrets has been to use a trusted third party, who is essentially a judge that can be called in to handle disputes between contract signers. There is also a large body of work following this approach; see, e.g., [ASW98] and references therein. Our solution follows the former paradigm.

Regarding work on "time," or relating time to computational effort, we mentioned the work of Dwork and Naor [DN92] on moderately-hard functions, which they used to combat junk e-mail. Timed primitives have also been designed for cryptographic key escrow. In [Sha95], Shamir suggested to escrow only partially a DES key, so that to recover the whole key, the government would need to first obtain the escrowed bits of the key, and then search exhaustively for the remaining bits. A problem with this type of approach is that the search can be parallelized, thus making it difficult to give a precise bound on the number of steps needed for the recovery. Bellare and Goldwasser [BG96,BG97] later suggested the notion of "time capsules" for key escrowing in order to deter widespread wiretapping, and where a major issue is the verification at escrow time that the right key will be recovered.

In another line of work, Rivest, Shamir and Wagner [RSW96] suggested "time-lock puzzles" for encrypting data, where the goal is to design puzzles that are "intrinsically sequential," and thus, putting computers to work together in parallel does not speed up finding the solution. They base their construction on a problem that, as we mentioned earlier, seems to satisfy that: modular exponentiation. In their work, however, no measures are taken to verify that the puzzle can be unlocked in the desired time.

Using a function similar to that of [RSW96]—specifically, the vector (1) above—Boneh and Naor [BN00] defined the notion of (verifiable) *timed commitments*, an extension to the standard notion of commitments in which a potential

[1] We observe that, as pointed out in [GJ02], these techniques are more general and can also be applied to other cryptographic functions, allowing, for example, the fair exchange of (verifiable) ciphertexts (e.g., by using ElGamal encryption); in this extended abstract we concentrate on the signatures application.

forced opening phase permits the receiver to recover (with effort) the committed value vithout the help of the committer. They show how to use timed commitments to improve a variety of applications involving time, including timed signatures of a special kind, and, in particular, contract signing. They show how to exchange Rabin and RSA signatures when the respective moduli coincide with the one used in vector (1). In [GJ02], Garay and Jakobsson show how to generate vector (1)-type structures very efficiently—they call these "derived time-lines," and use them together with blinding techniques for the timed release of standard signatures.

That is the work most closely related to ours. Further references to work on timed primitives and applications include [BDS98,GS98,Syv98].

Our work. Our contributions are two-fold:

- We achieve timed fair exchange (and contract signing) of standard signatures—specifically, of those which admit blinding (e.g., RSA, DSA, Schnorr). We achieve this in three steps. First, we extend the time structure (1) to what we call *mirrored time-lines*: points whose "distance" increases in exponential fashion, followed by points whose distance decreases like-wise. Second, using this new structure, we design a protocol for the timed fair exchange by two parties of arbitrary values (values lying on their respective mirrored time-lines). Finally, we apply the blinding techniques of [GJ02] to turn this protocol into a protocol for the timed fair exchange of standard signatures.
- Using a "layered safe prime" construction, we show how under reasonable assumptions, namely, the Hardy–Littlewood version of the prime k-tuples conjecture [HL23], the period of the underlying sequence can be made large enough so that cycling will not occur in a time-line whose modulus is the product of two such safe primes.

Organization of the paper. Section 2 contains the necessary background material for the rest of the paper. The notion of mirrored time-lines, together with considerations on their periods as well as a protocol to construct them, is presented in Section 3. Section 4 is devoted to timed fair exchange. First we present the protocol for the fair exchange of arbitrary values (Section 4.1), and then we show how to turn it into a protocol for the fair exchange of signatures (Section 4.2). We conclude with some remarks on the efficiency of our protocol.

2 Preliminaries

The generalized BBS assumption. Let N be a Blum integer, i.e., $N = p_1 p_2$, where p_1 and p_2 are distinct primes each congruent to 3 mod 4. Recall the notion of a Blum-Blum-Shub (BBS) sequence $x_0, x_1, \cdots, x_n$, with $x_0 = g^2 \pmod N$ for a random $g \in \mathbb{Z}_N$, and $x_i = {x_{i-1}}^2 \pmod N$, $1 \leq i \leq n$. It is shown in [BBS86] that the sequence defined by taking the least significant bit of the elements above is polynomial-time unpredictable (unpredictable to the left and

to the right), provided the quadratic residuosity assumption (QRA) holds. Recall also that these sequences are periodic (although not always purely periodic).

In [BN00], Boneh and Naor postulate the following generalization of unpredictability of BBS-type sequences. Let N and g be as above, and k an integer such that $l < k < u$. Then given the vector

$$< g^2, g^4, g^{16}, \ldots, g^{2^{2^i}}, \ldots, g^{2^{2^{k-1}}}, g^{2^{2^k}} > (\mathrm{mod} N), \tag{2}$$

the (l, u, δ, ϵ) generalized BBS assumption states that no PRAM algorithm whose running time is less than $\delta \cdot 2^k$ can distinguish the element $g^{2^{2^{k+1}}}$ from a random quadratic residue R^2, with probability larger than ϵ. The bound l precludes the parallelization of the computation, while the bound u the feasibility of computing square roots through factoring. We refer to [BN00] for further details on this assumption.

Time-lines. In [GJ02], Garay and Jakobsson called the partial description of the BBS sequence $\{g^{2^{2^i}}\}_{i=0}^{k}$ (mod N) as given by input vector (2) a *value-hiding time-line*, calling the value $g^{2^{2^{k-1}}}$ the time-line's *hidden value*. They defined the notion of a (T, t, ϵ) *time-line commitment*, which allows a committer to give the receiver a *timed* commitment to a hidden value. At a later time, she can reveal this value and prove that it's the correct one. However, in case the committer fails to reveal it, the receiver can spend time T and retrieve it.

More specifically, a (T, t, ϵ) time-line commitment consists of three phases: the **commit** phase, where the committer commits to the time-line's hidden value by giving the receiver the description of the time-line and a proof of its well-formedness; the **open** phase, where the committer reveals information to the receiver that allows him to compute the hidden value—and be convinced that this is indeed the committed value; and the **forced open** phase, which takes place when the committer refuses to execute the open phase; in this case, the receiver executes an algorithm that allows him to retrieve the hidden value, and produces a proof that this is indeed the value. A time-line commitment scheme must satisfy the following security constraints:

Binding: The value committed to is uniquely defined by the commitment. In particular, it is not possible to open up the commitment to a value other than that which will be obtained by the forced opening of the commitment (corresponding to the iterated squaring of the time-line's starting value.)

Soundness: After receiving the time-line, the receiver is convinced that the forced open phase will produce the hidden value in time T.

Privacy: Every PRAM algorithm whose running time is at most $t < T$ on polynomially many processors, given the transcript of the time-line commit protocol as input, will succeed in computing the time-line's hidden value with probability at most ϵ.

Proving the well-formedness of a time-line involves a computational effort (see [BN00], and our variant in Section 3). Garay and Jakobsson [GJ02] show

how once a time-line is established and its properties verified, time-line commitments can be generated at a low cost. At a high level, the idea is for the committer to create a new time-line from the original time-line by applying a secret transformation value—the *shifting* exponent—to the end points of the master time-line, in such a way that verification of the new time-line's properties is easy. We will be using time-line commitments in Section 4. Further details can be found in [GJ02].

3 Mirrored Time-Lines

Recall that, in a nutshell, a time-line is the partial description of a BBS sequence, as given by the vector (2) in Section 2. In this section we introduce a time structure that is an extension of a time-line. This new time structure will be used by our applications of Section 4.

Let N be a Blum integer as before (i.e., $N = p_1 p_2$, where p_1 and p_2 are distinct primes each congruent to 3 mod 4) and $g \in_{\mathrm{R}} \mathbb{Z}_N$. Let $u_i \stackrel{\text{def}}{=} g^{2^{2^i}} \bmod N$, $0 \le i \le K$, and $v_j \stackrel{\text{def}}{=} g^{2^{2^{K+1}-2^{K-j}}} \bmod N$, $1 \le j \le K$. We call a *mirrored time-line* (MTL) the result of "concatenating" two such vectors, and throwing in the initial term g, and the final term $u_{k+1} = g^{2^{2^{K+1}}} \bmod N$, i.e.:

$$\langle\, g, \underbrace{g^2, g^4, \cdots, g^{2^{2^{K-1}}}}_{\{u_i\}_{i=0}^{K-1}}, \underbrace{g^{2^{2^K}}}_{u_K = v_0}, \tag{3}$$

$$\underbrace{g^{2^{2^K+2^{K-1}}}, g^{2^{2^K+2^{K-1}+2^{K-2}}}, \cdots, g^{2^{2^K+2^{K-1}+\cdots+1}}}_{\{v_j\}_{j=1}^{K}}, \underbrace{g^{2^{2^{K+1}}}}_{u_{K+1}} \,\rangle$$

(everything mod N). In other words, in the first half of an MTL the distance between the points grows geometrically, as with original time-lines,while in the second half it *decresases* like-wise. Sometimes we will call element $u_K (= v_0)$ the *pivot* of the MTL.[2] Obviously, the length of an MTL is double that of a basic time-line—for the same K. In fact, our applications will require using larger values of K (e.g., $K \approx 80$) than the ones used by the timed applications of [BN00, GJ02] ($K \in [30, 50]$); thus, measures must be taken to guarantee that the period of the underlying sequence is large enough. In the following, we present a protocol for MTL generation between two parties, the prover (the party generating the MTL) and the verifier, with the following properties:

1. It assures the verifier that with high probability all the points in the received MTL lie on the same time-line; and

[2] The main application of the MTL, the fair exchange of signatures, will only be using the second half of the time-line; however, the first half of the time-line will be needed to prove the well-formedness of the second half (cf. Section 3.2).

2. it guarantees the prover that the period of the underlying sequence is larger than the length of the MTL.

As in [BN00,GJ02], property 1 can be achieved rather efficiently, with roughly $2K$ zero-knowledge proofs that can be run in parallel. Property 2, however, requires some additional considerations, which we outline below.

3.1 Obtaining Large Periods

Given a nonzero integer g and a positive integer n let $\mathrm{Per}(g,n)$ be the period of the (ultimately) periodic sequence $(g^i \bmod n)_{i\ge 0}$, let $\mathrm{Per}_1(g,n)$ be the period of $(g^{2^i} \bmod n)_{i\ge 0}$, and let $\mathrm{Per}_2(g,n)$ be the period of $(g^{2^{2^i}} \bmod n)_{i\ge 0}$. We have

$$\mathrm{Per}_2(g,n) = \mathrm{Per}(2,\mathrm{Per}_1(g,n)) = \mathrm{Per}(2,\mathrm{Per}(2,\mathrm{Per}(g,n))). \tag{4}$$

Since $\mathrm{Per}(g,n)$ is smaller than n, and in fact can be much smaller, it is not clear that a multiple layering of the Per function is adding complexity. In fact it may be that the reverse is occurring, and a sequence $(g^{2^{2^i}} \bmod n)_{i\ge 0}$ has a very short period. Thus, in using a time-line sequence as a means of imposing a given degree of computational intractability, it seems interesting to know that the period of such a sequence is suitably large.

If g and n are coprime integers, with $n > 0$, let $\mathrm{ord}(g,n)$ be the multiplicative order of g in $\mathbb{Z}_n^*$. More generally, even when g and n are not coprime, let $\mathrm{ord}^*(g,n)$ be $\mathrm{ord}(g,n^*)$, where n^* is the largest divisor of n that is coprime to g. The following result is well-known.

Lemma 1. *For nonzero integers g,n with $n>0$, $\mathrm{Per}(g,n) = \mathrm{ord}^*(g,n)$.*

Proof. Write $n = n_0 n^*$. By the Chinese remainder theorem, our sequence $(g^i \bmod n)_{i\ge 0}$ is equivalent to the sequence of pairs $(g^i \bmod n_0, g^i \bmod n^*)_{i\ge 0}$. For i large, $g^i \equiv 0 \bmod n_0$, so it has an ultimate period of length 1. Further, $(g^i \bmod n^*)_{i\ge 0}$ is purely periodic with period $\mathrm{ord}(g,n^*)$. Thus $(g^i \bmod n)_{i\ge 0}$ becomes periodic as soon as $g^i \equiv 0 \bmod n_0$, and the length of the period is $\mathrm{ord}^*(g,n)$. ∎

We have at least some lower bound before a repeat in the sequence $(2^i \bmod n)_{i\ge 0}$. Indeed, if $2^j \equiv 2^i \bmod n$ for $j > i \ge 0$, then $j > \log n$. It follows then from (4) that the first $\lfloor \log\log(\mathrm{Per}(g,n)) \rfloor$ terms of the sequence $(g^{2^{2^i}})_{i\ge 0}$ are distinct.

On the other hand, since $\mathrm{Per}_2(g,n) < \mathrm{Per}_1(g,n) < \mathrm{Per}(g,n)$ (from (4) and Lemma 1) if we have g,n with $\mathrm{Per}_2(g,n)$ large, it follows that $\mathrm{Per}_1(g,n)$ and $\mathrm{Per}(g,n)$ are large as well.

Here are some possible strategies for choosing a pair g,n with $\mathrm{Per}_2(g,n)$ large:

- Actually compute $\mathrm{Per}_2(g,n)$ for randomly chosen parameters g,n (with n a Blum integer) to see if it is long enough.

- Ignore the problem, hoping that a randomly chosen g, n will do.
- Use *layered* safe primes as described below.
- Use a combination of the above strategies; that is, use safe primes that are not layered, or layered to only the next level.

Computing the period for randomly chosen parameters may in fact be difficult, depending on one's ability to factor various numbers that arise. In some sense, using layered safe primes allows for the computation of the period in direct fashion.

Ignoring the problem, which has been the tacit strategy till now, would work in practice if there are not many pairs g, n where $\mathrm{Per}_2(g, n)$ is very small. Such a result was shown in [FPS01a,FPS01b] for the sequence $(g^{h^k} \bmod n)_{k\geq 0}$, where g, h, n are chosen at random, with n the product of 2 primes of the same magnitude. A somewhat weaker, but still adequate result is shown in the same papers for the special case $h = 2$. It seems likely that for most Blum integers n and for most residues $g \bmod n$, $\mathrm{Per}_2(g, n) > n^{1-\epsilon}$ for any fixed $\epsilon > 0$, but this remains unproved at present. A paper which takes a step in this direction is [MP03].

A prime p is "safe" if $(p-1)/2$ is also prime. (In number theory, a prime q with $2q + 1 = p$ also prime is known as a Sophie Germain prime.) It has long been recommended to use safe primes in RSA moduli in order to foil the $p - 1$ factoring algorithm. However, almost surely this algorithm would be useless with random primes in an RSA modulus (see [PS95]). Despite this, since it is so easy to use safe primes, the thought then is "Why not? Let's use safe primes." The same logic can be applied to time-line sequences.

Consider integers s such that

$$s, \quad r = 2s + 1, \quad q = 2r + 1, \quad \text{and} \quad p = 2q + 1$$

are all prime. By the Hardy–Littlewood version of the prime k-tuples conjecture [HL23], the number of such integers $s \leq x$ is

$$\sim cx/(\ln x)^4,$$

as $x \to \infty$, where

$$c = \frac{2}{3} \prod_{p \text{ prime}} \left(1 - \frac{6p^2 - 4p + 1}{(p-1)^4}\right) \approx 5.5349.$$

While it is perhaps not so easy to find large examples of such numbers s, it is not intractable. For example, in the vicinity of 2^{500}, about 1 in 26 billion integers will (conjecturally) be valid examples. A sieve may be used to isolate numbers s which are more likely to work, with a primality test to finish the job. If successful values s_1, s_2 are found, one can construct the modulus $N = p_1 p_2$ with these layered safe primes. Doing so ensures that if $\gcd(g^3 - g, N) = 1$ then the period of the sequence

$$g^{2^{2^i}} \bmod N, \quad i = 0, 1, \ldots, \tag{5}$$

is either $s_1 s_2$ or $2s_1 s_2$, where $s_1 s_2 \approx 2^{-6}N$. Indeed, the condition $\gcd(g^3 - g, N) = 1$ implies that g is coprime to N and that $\mathrm{ord}(g, p_i) > 2$ for $i = 1, 2$. Thus, $q_1 q_2$ divides $\mathrm{ord}(g, N)$. Similarly, $r_1 r_2$ divides $\mathrm{ord}(2, q_1 q_2)$ and $s_1 s_2$ divides $\mathrm{ord}(2, r_1 r_2)$. It follows from Lemma 1 and (4) that $s_1 s_2$ divides $\mathrm{Per}_2(g, N)$.

We thus have the following theorem.

Theorem 1. *Assuming the Hardy–Littlewood version of the prime k-tuples conjecture there are* $\sim 1.4986 \cdot 2^m/(m-3)^4$ *safe primes p with m bits such that* $(p-1)/2 = q$ *and* $(q-1)/2 = r$ *are also safe primes. Further, if N is the product of any 2 of these primes p and g is any integer satisfying* $\gcd(g^3 - g, N) = 1$, *then the period of the sequence (5) is at least* 2^{2m-8}.

As noted above, having $\mathrm{Per}_2(g, N)$ large ensures that $\mathrm{Per}_1(g, N)$ is also large, since $\mathrm{Per}_1(g, N) > \mathrm{Per}_2(g, N)$. This then ensures that the points u_i, v_j on an MTL under consideration are all distinct. Indeed, if r_1, r_2 have 500 or more bits each (to make N hard to factor), and if $K \approx 80$ as recommended, we are only looking at a subset of the first $2^{\approx 80}$ terms of a sequence with period more than 2^{998}.

A less "high tech" way is likely to work to find Blum integers N where the period of (5) is large. Suppose q_1, q_2 are random primes near 2^{500}. In [FPS01a, FPS01b] it is shown that for random choices of g, h it is highly likely that the period of the sequence

$$g^{h^i} \bmod q_1 q_2, \quad i = 0, 1, \ldots,$$

is greater than $(q_1 q_2)^{1-\epsilon}$. It is reasonable to conjecture that the same is true for the sequence

$$2^{2^i} \bmod q_1 q_2,$$

and that this remains true if in addition we insist that both $p_1 = 2q_1 + 1, p_2 = 2q_2 + 1$ are prime. Assuming this is so, it is an easy matter then to select random primes q_1, q_2 near 2^{500} such that

(1) $p_1 = 2q_1 + 1$, $p_2 = 2q_2 + 1$ are prime, and

(2) the period of the sequence $2^i \bmod q_1 q_2$ exceeds 2^{900}.

(One can either hope that property (2) is satisfied without checking it, or one can choose the primes q_1, q_2 such that $\phi(q_1 q_2)$ is easy to factor, so that the period of the sequence might actually be checked.) Then, if $N = p_1 p_2$ and g is chosen with $\gcd(g^3 - g, N) = 1$, then $\mathrm{Per}_1(g, N) > 2^{900}$ and the first 900 terms of the sequence (5) are distinct. Both conditions give a quite comfortable margin to avoid periodicities.

In what follows, we take this last approach of using safe primes p_1, p_2 for N, with $p_i = 2q_i + 1$ for $i = 1, 2$, and with $\mathrm{ord}(2, q_1 q_2) > 2^{900}$.

3.2 The Mirrored Time-Line Protocol

We now outline the steps of the MTL protocol.

1. **Setup.** The prover chooses a modulus $N = p_1 p_2$, for p_1, p_2 safe m-bit primes as in Theorem 1. The prover then proves to the verifier in zero-knowledge that N is the product of two safe primes, using the protocol of Camenisch and Michels [CM99a]. The prover also chooses a generator g with $g^3 - g \in \mathbb{Z}_N^*$. Then the order of g in $\mathbb{Z}_N^*$ is either $2q_1q_2$ or q_1q_2, and the order of g^2 is q_1q_2. The condition that $g^3 - g \in \mathbb{Z}_N^*$ is easily checked by both the prover and the verifier.

2. **Compute the MTL.** The prover computes the elements in the MTL, $u_i = g^{2^{2^i}} \bmod N$, $0 \leq i \leq K$, and $v_j = g^{2^{2^{K+1}-2^{K-j}}} \bmod N$, $1 \leq j \leq K$. Knowing $\phi(N)$ allows the prover to perform the computations efficiently, by first computing, e.g., $a_i = 2^{2^i} \bmod \phi(N)$ and then $u_i = g^{a_i} \bmod N$. The computation of the v_i's is performed similarly.

3. **Prove well-formedness.** The prover proves to the verifier that $u_i = g^{2^{2^i}} \bmod N$, for $0 \leq i \leq K+1$, and $v_j = g^{2^{2^{K+1}-2^{K-j}}} \bmod N$, for $1 \leq j \leq K$. The proof for the u_i's is done as in [BN00], by showing that each triple $< g, u_i, u_{i+1} >$, $0 \leq i < K$, is of the form $< g, g^x, g^{x^2} >$, for some x.[3] Similarly, the correctness of the "v line" is established by a zero-knowledge proof that the tuples $< g, u_{K-1}, u_K, v_1 >$, and $< g, u_{K-j}, v_{j-1}, v_j >$, for $2 \leq j \leq K$, are Diffie-Hellman tuples. These $2K$ proofs are performed in parallel.

We now argue how the properties mentioned at the beginning of the section are satisfied upon completion of the protocol.

1. The probability that a cheating prover successfully convinces the verifier that a wrong value lies on the MTL is $2K\varepsilon_s$, where ϵ_s is the soundness error of the zero-knowledge protocol of step 3. Given the safe-prime structure of N, and using the ZK protocol presented in [Mao01] yields

$$\varepsilon_s = \frac{2q_1 + 2q_2 - 1}{2q_1q_2} \approx \frac{4}{\sqrt{N}}.$$

   These protocols can be repeated to lower the error.

2. A sufficiently large period of the MTL follows from Theorem 1.

Setting up an MTL requires a computational effort, particularly in the Setup phase for the proof of the safe-prime structure of N [CM99a]. In our applications of Section 4, however, this effort will be incurred only once, as the prover will use one MTL (call it the *master* MTL) to spawn many other MTL's much more efficiently. We also note the "zero-knowledgeness" of the MTL protocol above with respect to $\phi(N)$; this will guarantee the timed-release property of the applications that follow.

[3] See also [Mao01] for a more efficient protocol.

4 Timed Fair Exchange of Standard Signatures

We now turn to applications of our MTL construction. One important application is fair exchange. Recall that in this problem two parties, Alice and Bob, have an item that they wish to exchange in such a way that either both get the other's item, or nobody does. In particular, we will be considering the case of exchange of digital signatures, a specific instantiation of which is *contract signing*, where each party would like to receive the other's signature on a known piece of text. But before we do that, we first consider the simpler case of the parties fairly exchanging their respective time-lines' hidden values (see Section 2). We will then use this building block to allow for the exchange of standard signatures, by using the committed hidden values to blind the respective signatures—and prove that this is indeed the case. Note that the simpler building block already enables other timed applications, such as collective (two-party) coin tossing [Blu81].

4.1 Fair Exchange of Hidden Values

Assume that the two parties, Alice and Bob, have agreed on an integer parameter K. Let $T = 2^K$. For this application, we envision K to be large (in particular, larger than for previous timed applications), e.g., $K \approx 80$. Further, assume that both parties have performed (T, t, ϵ) time-line commitments to the other party, using the protocol of [GJ02]. As a result, Alice and Bob are committed to their respective time-lines' hidden values which now they would like to exchange fairly. We adapt the definition of fairness of a timed contract signing protocol of [BN00] to the case of exchange of hidden values:

Fairness: A hidden-value fair exchange protocol is said to be (a, ϵ)-*fair* if for any of the parties (wlog, Alice) working in time t smaller than some security parameter, and running the fair exchange protocol with the other party (Bob), the following holds: If at some point Alice aborts the protocol and succeeds in recovering Bob's hidden value with probability p_A, then Bob running in time $a \cdot t$ can recover Alice's hidden value with probability p_B, such that $|p_A - p_B| \leq \epsilon$.

Intuitively, a is the *advantage* of the party that aborts. Below we present a hidden-value fair exchange protocol that is $(2, \epsilon)$-fair, for negligible ϵ. The general idea is as follows. First, the parties each generate an MTL as described in Section 3.[4] To these MTL's the parties apply the time-line commitment protocol of [GJ02]; as a result, new, *derived* time-lines are generated (much more efficiently), and the parties become committed to the new time-lines' hidden values. It turns out that a much shorter description (only three points) of these

[4] This phase of the protocol has to be performed only once, and then the resulting time-lines can be used repeatedly for several exchanges. It is also possible for both parties to use the same mirrored time-line, say, a "public" mirrored time-line generated by a third party which each party verifies. For simplicity, we assume that each party generates its own.

new time-lines is sufficient for the commitment protocol: the end points and the new time-line's pivot; the hidden value is the (principal) square root of the right end point, and remains secret. Having established this, the parties proceed to perform the exchange: starting from the time-line's pivot, the parties now take turns, revealing and verifying the v points in ascending order (i.e., from left to right) on the new time-line., Thus, after K rounds, they both reach and verify each other's hidden values. Should any of the parties abort in the middle of the exchange, the other party is assured that the hidden value can be reached by repeated squaring of the last revealed value. Note that it is possible for a party to abort after receiving the first party's commitment and without having sent his own; given our choice of parameters K and T, however, the amount of work—and time—required to get to the first party's hidden value will be gargantuan. We now describe the protocol in detail.

1. **Setup.** Both Alice and Bob generate their own *mirrored* time-line, MTL_A and MTL_B respectively, using the protocol from Section 3, which gets verified by the other party.

2. **Commit phase.** Both parties, acting as committers, execute the time-line commitment protocol of [GJ02] using their respective MTL's as the *master* time-line. Specifically, each party $i \in \{\text{Alice, Bob}\}$ acting as the committer, performs the following steps on the mirrored time-line he generated (when necessary, we will be using superscripts to identify the elements of each party's time-line; in the following we omit them for simplicity):

   2.1 *Choose shifting exponent.* The committer picks a random integer $\alpha \in \mathbb{Z}_{\phi(N)}$. (If the committer does not know the value of $\phi(N)$ because of use of an MTL from a trusted third party, then the committer may choose α at random in $[1, N/2]$.)

   2.2 *Compute new time-line.* The committer computes $g' = g^\alpha$, ${u'}_K = u_K^\alpha$, ${v'}_K = v_K^\alpha$, and ${u'}_{K+1} = u_{K+1}^\alpha \pmod N$. He outputs $\langle g', {u'}_K, {u'}_{K+1}\rangle$, and **keeps** ${v'}_K$ **secret**. ${v'}_K$ is party i's hidden value.

   2.3 *Prove well-formedness of new time-line.* The committer does not reveal α, but instead proves to the receiver that

   $$\log_g g' = \log_{u_K} {u'}_K = \log_{u_{K+1}} {u'}_{K+1} (= \alpha),$$

   i.e., that the new time-line is correctly derived from the mirrored time-line, using a (statistical) zero-knowledge proof of equality of two discrete logs modulo the same number (N) [CEvdG87,CP92, Bao98]. Call this type of proof EQLOG-1(α, N). (For conciseness, we will sometimes use "$[\cdot]$" to refer to these zero-knowledge proofs.)

3. **Exchange phase.** Now the gradual exchange starts, with Alice and Bob taking turns in revealing their respective v''s in ascending order (i.e., halving the distance to the hidden value in each round), together with a proof

that they lie on the shifted time-line. Specifically, the following is being executed from $j = 1$ to K: Alice sends to Bob

$$v'^{A}_{j} = (v^{A}_{j})^{\alpha} \bmod N_A, \quad [\log_{g^A} g'^{A} = \log_{v^A_j} v'^{A}_{j}],$$

using an EQLOG-1 proof as before. Bob, after verifying the proof, responds with

$$v'^{B}_{j} = (v^{B}_{j})^{\beta} \bmod N_B, \quad [\log_{g^B} g'^{B} = \log_{v^B_j} v'^{B}_{j}].$$

4. **Forced retrieval.** Should any of the parties, say, Alice, stop the exchange at round ℓ, $1 \leq \ell < K$, then Bob proceeds to successively square $v'^{A}_{\ell-1}$ $(\mathrm{mod}\, N_A)$ to get to v^A_K, Alice's hidden value, in approximately $2^{K-\ell}$ modular multiplications.

We now argue for the security of the protocol. The well-formedness of the derived time-line in Step 2 follows from the well-formedness of the MTL construction (Section 3) and the proofs of correct shifting, both in Step 2 and in the gradual exchange of Step 3. Thus, the parties are assured that the received points lie on the new time-line. Additionally, the new time-line inherits the long-period guarantee of the original MTL; thus, assuming the generalized BBS assumption holds, no "shortcuts" are possible for any adversary working in time $t < T$, and the probability of obtaining the other party's hidden value is at most ϵ. For the fairness, suppose, wlog, that Alice aborts the protocol after Bob reveals only $\ell < K$ points v' on his time-line. Then she can compute the hidden value in approximately $2^{K-\ell}$ multiplications (but not faster, based on the generalized BBS assumption). Bob, on the other hand, has one v' less than Alice has, and can compute Alice's hidden value using approximately $2 \cdot 2^{K-\ell} = 2^{K-\ell+1}$ modular multiplications; thus, his workload is roughly twice that of Alice's. Thus we have the following.

Lemma 2. *Assume the hardness of the discrete logarithm problem, the Hardy–Littlewood version of the prime k-tuples conjecture, and that the generalized BBS assumption holds for some parameters* (l, u, δ, ϵ)*. Then the protocol above is a* $(2, \epsilon)$*-fair hidden-value exchange protocol.*

As in [BN00], it is possible to argue the "zero-knowledgness" of a party's hidden value against an adversary willing to invest a time less than T, by constructing a simulator operating in time proportional to the running time of the adversary that outputs a transcript of the execution that is indistinguishable from the real execution. This property will additionally guarantee the feature termed "abuse freeness" in [GJM99] (aka "strong fairness" [BN00]) in contract signing protocols, which we now consider.

4.2 Fair Exchange of Signatures

In [GJ02], Garay and Jakobsson show how to use time-line commitments to perform the timed release of signatures that allow for "blinding" [Cha82] (e.g.,

RSA, DSA, Schnorr). Timed release of signatures is closely related to our timed fair exchange problem, except that it is only "one way," from a signer to a verifier: the signer time-commits to a signature, and the verifier knows that if the signer refuses to reveal the signature, he will be able to retrieve it using a "forced-open" phase. In [GJ02] the time-line's hidden value is used as the blinding factor for the signature—together with proofs that, once recovered, the hidden value will produce correct unblinding of the signature. By applying this transformation to our hidden-value fair exchange protocol, we obtain a fair exchange protocol for standard signatures.

Note that the **Fairness** condition of Section 4.1 can be readily modified to account for the fair exchange of signatures. Two other standard conditions for the case of exchange of signatures are **Completeness** (the signature verification algorithms will output "Accept" on signatures that are the result of correct execution of the protocol), and **Unforgeability** (the probability that any polynomial-time adversary will produce a fake signature/contract is negligible); see, e.g., [ASW98,BN00] for further details on these and other definitions.

We now show how to augment the Commit phase of Section 4.1 to perform the signature blinding. We exemplify for the case of RSA signatures,[5] but keep in mind that the technique also applies to other (e.g., discrete log-based) signatures; also note that the signatures being exchanged could be of different types. The transformation is taken from [GJ02] almost *verbatim* (with some details omitted for readability). We assume that steps 2.1–2.3 regarding the time-line commitment have already taken place, and, again, we omit indices identifying the parties for readibility, but bear in mind that both parties are performing this phase, as signers.

2. **Commit$^+$ phase.** Let n be the signer's RSA modulus, (e, n) be the signer's public key, and d his secret key, chosen in such that way that $x^{ed} = x \bmod n$ for all values $x \in \mathbb{Z}_n$. Additionally, the signer will be performing auxiliary (standard) commitments using a subgroup of order N in $\mathbb{Z}^*_{N'}$, for $N' = \kappa N + 1$ a prime; let h be a generator of this subgroup.

   2.4 *Normal signature generation.* Let M be the message to be signed.[6] The signer computes $s = M^d \bmod n$.

   2.5 *Application of blinding factor.* The signer blinds the signature by computing

   $$\tilde{s} = s^{1/v_K} \bmod n,$$

   where v_K is his time-line's hidden value. He sends the pair $(M, \tilde{s})$ to the verifier.

[5] Timed contract signing using RSA (and Rabin) signatures was also discussed in [BN00], but for the case when the same modulus is used for both the signature and the time-line construction. Here we consider the general case of arbitrary moduli.

[6] For simplicity, we omit here the issue of secure padding schemes for digital signatures.

2.6 *Auxiliary commitments and proof of uniqueness.* The signer computes

$$b = h^{v_K} \quad \text{and} \quad B = h^{u_{K+1}} \pmod{N'}.$$

The signer proves to the verifier that $\log_h b = \log_b B\ (= v_K)$, using EQLOG-1$(v_K, N')$.

Let INTVL$(x \in [a, b])$ denote a zero-knowledge proof of knowledge that x lies exactly (i.e., with expansion rate 1) in interval $[a, b]$ [Bou00]. The signer proves to the verifier that the blinding factor lies in the right interval with INTVL$(v_K \in [0, N-1])$.

2.7 *Proof of correct blinding.* The verifier computes $X = \tilde{s}^e \bmod n$.

Let EQLOG-2(x, n_1, n_2) denote a zero-knowledge proof of knowledge of equality of two discrete logs (x) in *different* moduli (n_1 and n_2) [BT99,CM99b]; these proofs additionally require the strong RSA assumption [BP97]. The signer proves that

$$\log_X (M \bmod n) = \log_h (b \bmod N')\ (= v_K)$$

using EQLOG-2(v_K, n, N').

It is shown in [GJ02] that the above sub-protocol produces a (T, t, ϵ) timed RSA signature scheme. This, together with Lemma 2, allow us to conclude

Theorem 2. *Assume the hardness of the discrete logarithm problem, the Hardy–Littlewood version of the prime k-tuples conjecture, and that the strong RSA assumption and the generalized BBS assumption hold, the latter for some parameters (l, u, δ, ϵ). Let $\mathcal{S} = (G, S, V)$ be a signature scheme that allows for blinding (e.g., RSA, DSA, Schnorr). Then it is possible to construct protocols for the timed fair exchange of signatures based on $\mathcal{S}$.*

4.3 Efficiency

Regarding efficiency, the number of rounds of our protocol is roughly the same as the protocol of [BN00] for special signatures (K). The computational cost of our protocol in each round is higher, though: generation and verification of a zero-knowledge proof of knowledge, vs. a simpler verification operation (e.g., a squaring in the case of Rabin signatures) in the case of [BN00]. On the other hand, the setup cost is lower, as the cost involved in the generation of the mirrored time-line is incurred only once, after which derived time-lines are generated for each execution of the protocol, which require a constant number of proofs of knowledge.

Acknowledgements. The authors thank Markus Jakobsson for valuable discussions on the subject, and the anonymous reviewers for FC'03 for their useful comments.

References

[ASW98] N. Asokan, V. Shoup, and M. Waidner. Fair exchange of digital signatures. *Advances in Cryptology—EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606, Springer-Verlag.

[Bao98] F. Bao. An efficient verifiable encryption scheme for encryption of discrete logarithms. In *Proc. CARDIS'98*, 1998.

[Ble00] D. Bleichenbacher. On the distribution of DSA session keys. Manuscript, 2000.

[Blu81] M. Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *Advances in Cryptology—CRYPTO '81*, pages 11–15. ECE Report 82-04, 1982.

[Blu83] M. Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1(2):175–193, May 1983.

[BBS86] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing*, 15(2):364–383, May 1986.

[BCDvdG87] E. Brickell, D. Chaum, I. Damgård, and J. van de Graaf. Gradual and verifiable release of a secret (extended abstract). In *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 156–166, Springer-Verlag, 1988.

[BDS98] M. Burmester, Y. Desmedt and J. Seberry. Equitable Key Escrow with Limited Time Span. In *Advances in Cryptology—Asiacrypt '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 380–391, Springer-Verlag, 1998.

[BG96] M. Bellare and S. Goldwasser. Encapsulated key escrow. In MIT/LCS/TR-688, 1996.

[BG97] M. Bellare and S. Goldwasser. Verifiable partial key escrow. In *Proc. ACM CCS*, pages 78–91, 1997.

[BN00] D. Boneh and M. Naor. Timed commitments (extended abstract). In *Advances in Cryptology—CRYPTO '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 236–254, Springer-Verlag, 2000.

[Bou00] F. Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology—EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444, Springer-Verlag, 2000.

[BP97] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology–Eurocrypt '97*, pages 480–494, 1997.

[BT99] F. Boudot and J. Traoré. Efficient publicly verifiable secret sharing schemes with fast or delayed recovery. In *Proc. 2nd International Conference on Information and Communication Security*, volume 1726 of *Lecture Notes in Computer Science*, pages 87–102, Springer-Verlag, 1999.

[Cha82] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, pages 199–203. Plenum Press, New York and London, 1983.

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology—CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Springer-Verlag, 1994

[CEvdG87] D. Chaum, J. Evertse, and J. van de Graaf. An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In *Advances in Cryptology—EUROCRYPT 87*, volume 304 of *Lecture Notes in Computer Science*, pages 127–141, Springer-Verlag, 1988.

[CFT98] A. Chan, Y. Frankel, and Y. Thiounis. Easy come – easy go divisible cash. In *Advances in Cryptology—EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 561–575, Springer-Verlag, 1998.

[CM99a] J. Camenisch and M. Michels. Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes. In *Advances in Cryptology - EUROCRYPT '99*, . volume 1592 of *Lecture Notes in Computer Science*, pages 106-121, Springer Verlag, 1999.

[CM99b] J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes (extended abstract). In *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 414–430, Springer-Verlag, 1999.

[CP92] D. Chaum and T. Pedersen. Wallet databases with observers (extended abstract). In CRYPTO'92 [CRY92], pages 89–105.

[CRY92] *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, Springer-Verlag, 1993.

[Dam95] I. B. Damgård. Practical and provably secure release of a secret and exchange of signatures. *J. of Crypt.*, 8(4):201–222, Autumn 1995.

[DN92] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In CRYPTO'92 [CRY92], pages 139–147.

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, June 1985.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Springer-Verlag, 1987.

[FPS01a] J. B. Friedlander, C. Pomerance, and I. E. Shparlinski. Period of the power generator and small values of Carmichael's function. *Math. Comp.* **70** (2001), 1591–1605.

[FPS01b] J. B. Friedlander, C. Pomerance, and I. E. Shparlinski. Small values of the Carmichael function and cryptographic applications. In *Progress in Computer Science and Applied Logic*, Vol. 20, pages 25–32, Birkhäuser Verlag, Basel, Switzerland, 2001.

[Gol83] O. Goldreich. A simple protocol for signing contracts. In *Advances in Cryptology—CRYPTO '83*, pages 133–136.

[GJ02] J. Garay and M. Jakobsson. Timed Release of Standard Digital Signatures. In *Financial Cryptography '02*, volume 2357 of *Lecture Notes in Computer Science*, pages 168–182, Springer-Verlag, 2002.

[GJM99] J. Garay, M. Jakobsson and P. MacKenzie. Abuse-free Optimistic Contract Signing. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science* Springer-Verlag, pages 449-466, 1999.

[GMP01] S. Galbraith, W. Mao, and K. Paterson. A cautionary note regarding cryptographic protocols based on composite integers. In HPL-2001-284, 2001.

[GS98] D. Goldschlag and S. Stubblebine. Publicly Verifiable Lotteries: Applications of Delaying Functions. In *Financial Cryptography '98* volume 1465 of *Lecture Notes in Computer Science*, Springer-Verlag, 1998.

[HL23] G. H. Hardy and J. E. Littlewood, Some problems in "Partitio Numerorum," III: On the expression of a number as a sum of primes. *Acta Math.* **44** (1923), 1–70.

[Mao98] W. Mao. Guaranteed correct sharing of integer factorization with off-line shareholders. In *Proc. Public Key Cryptography '98*, pages 27–42, 1998.

[Mao01] W. Mao. Timed-Release Cryptography. In *Selected Areas in Cryptography VIII (SAC'01)*, volume 2259 of *Lecture Notes in Computer Science*, pages 342-357, Springer-Verlag, 2001.

[MP03] G. Martin and C. Pomerance. The normal order of iterates of the Carmichael λ-function. in progress.

[May93] T. May. Timed-release crypto. In `http://www.hks.net.cpunks/cpunks-0/1460.html`, 1993.

[PS95] C. Pomerance and J. Sorenson. Counting the integers factorable via cyclotomic methods. *J. Algorithms* 19: 250–265, 1995.

[RSW96] R. Rivest, A. Shamir, and D. Wagner. Time-lock puzzles and timed-release crypto. In MIT/LCS/TR-684, 1996.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Springer-Verlag, 1985.

[Sha95] A. Shamir. Partial key escrow: A new approach to software key escrow. In *Key Escrow Conference*, 1995.

[Syv98] P. Syverson. Weakly Secret Bit Commitment: Applications to Lotteries and Fair Exchange. In *Proceedings of the 1998 IEEE Computer Security Foundations Workshop (CSFW11)*, Rockport Massachusetts, June 1998.

Asynchronous Optimistic Fair Exchange Based on Revocable Items

Holger Vogt

Department of Computer Science
Darmstadt University of Technology
D-64283 Darmstadt, Germany
hvogt@cdc.informatik.tu-darmstadt.de

Abstract. We study the benefits of revocable items (like electronic payments which can be undone by the bank) for the design of efficient fair exchange protocols. We exploit revocability to construct a new optimistic fair exchange protocol that even works with asynchronous communication channels. All previous protocols with comparable properties follow the idea of Asokan's exchange protocol for two generatable items [Aso98, ASW98]. But compared to that, our protocol is more efficient: We need less messages in the faultless case and our conflict resolution is less complicated. Furthermore, we show that the generatability, which is required by [Aso98, ASW98], is difficult to implement in the context of some electronic payments. Instead, revocability of payments may be much easier to realize. Thus, our new protocol is very well suited for the fair exchange of revocable payments for digital goods.

1 Introduction

The exchange of valuable items like electronic payments and digital goods exposes the participating parties to the risk that the Internet connection is accidentally or even maliciously disrupted. This may lead to an unfair situation, if one party received the item while the other one did not get anything. A malicious party can simply disrupt the communication after it obtained the item from the other party and refuse to send its own item. To overcome these problems fair exchange protocols have been proposed. These protocols ensure that either both parties receive what they expected or nobody gains anything valuable. However, this kind of fairness has to be realized with a trusted third party (also called trustee or TTP), as protocols without a TTP (e.g. [EGL82, BOGMR90, Jak95, BN00]) cannot guarantee this strong notion of fairness [EY80].

A simple fair exchange solution is an active TTP that is involved in every exchange (e.g. [BP90, Tyg96, CHTY96, ZG96, FR97]). Much more efficient are *optimistic protocols* [ASW97], in which the TTP only has to participate if a conflict has to be resolved. Under the assumption that errors occur rarely these optimistic protocols minimize the number of requests to the TTP and thus prevent that it becomes a bottleneck during an exchange. But optimistic

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 208–222, 2003.
© Springer-Verlag Berlin Heidelberg 2003

fair exchange requires items that have special properties, namely generatability or revocability [ASW97]. The TTP must be able to either generate or revoke an item during conflict resolution. For this conflict resolution either synchronous or asynchronous communication with the TTP is assumed. As asynchronous communication is much more realistic, we only consider protocols which work with asynchronous communication.

The research on asynchronous optimistic fair exchange focuses on protocols that require generatable items (e.g. [ASW98, ZDB99, GJM99, ASW00, MS01]), and thus no optimistic fair exchange protocol for revocable items has been developed in the asynchronous system model yet.

In this paper we point out some drawbacks of generatable electronic payments and argue that revocability is often easier to implement. We present a new asynchronous optimistic fair exchange protocol that exploits revocability. We prove that it ensures the same fairness properties as Asokan's protocol for generatable items [Aso98, ASW98], which is generally considered the best currently known optimistic protocol. Especially for the fair purchase of digital goods our new protocol has some advantages over Asokan's protocol: The required number of messages in the faultless case is less compared to Asokan's protocol, and our conflict resolution protocols are less complicated.

The remainder of the paper is structured as follows: Section 2 states the system assumptions and defines the fairness level that we want to achieve. The revocability and generatability properties of the items are also introduced in this section. In Section 3 we discuss the deficiencies of previous exchange protocols for revocable items and argue that Asokan's protocol for generatable items leads to some problems as soon as generatable payments are considered. A solution to these problems is our new protocol for the exchange of a strongly revocable and a weakly generatable item, which we present in Section 4. We prove that this protocol has the same fairness properties as Asokan's protocol and show that it is even more efficient. Finally, we conclude this paper in Section 5.

2 Definitions

We first recall the standard system assumptions for fair exchange based on [Aso98] and define the desired properties of a fair exchange protocol. Then in Section 2.3 we introduce the notions of generatability and revocability that can be exploited by optimistic fair exchange protocols.

2.1 System Assumptions

In a fair exchange protocol two parties A and B want to exchange their items i_A and i_B. Both parties possess a detailed description of the item they want to receive during this exchange. We use $desc_{i_A}$ for B's description of i_A, while $desc_{i_B}$ describes i_B. We assume that a verification function is available that takes an item and its description and outputs *true* if and only if the item satisfies the properties specified in the description. For example, if A wants to check a

received item i_B, he has to test whether the result of the verification function $check(i_B, desc_{i_B})$ is *true*.

There is a trusted third party (TTP), which can be contacted in the case of a detected conflict. The TTP is always available and processes every request atomically.

The communication channels to and from the TTP are *asynchronous*, which means that messages are delivered after an arbitrary but finite amount of time. As no messages are lost, this guarantees a response from the TTP. In contrast the channel between A and B may lose messages or might even be disrupted permanently by one of the parties.

A different communication model is called *synchronous*, as it assumes synchronous communication to the TTP which ensures message delivery within a fixed time. In [ASW00, Section 2.2] it is argued that these synchronous protocols (e.g. [ASW97, XYZZ99, Sch00]) do not make sense in a practical scenario with rather unreliable Internet communication. Due to this reason we restrict ourselves to *asynchronous protocols*, which work with the asynchronous communication channels defined above.

For simplicity, we assume that all channels are confidential and that the integrity and authenticity of messages is ensured. Furthermore, unique transaction identifiers should be incorporated in every message to prevent replay attacks. These transaction identifiers should be computed by hashing the descriptions of the items and all the other information about this exchange. Then it is always possible to link a message to its exchange transaction. Nevertheless, we omit these identifiers in our protocol descriptions in order to ease presentation.

2.2 Protocol Properties

A protocol is called a fair exchange protocol, if it fulfills the following requirements. Some properties are formulated with respect to one of the parties which is simply called P. Then the other party will be named Q.

Effectiveness: If no messages are lost, both parties behave according to the protocol and do not want to abandon the exchange, then both parties receive the desired items.

Fairness for P: If P behaves according to the protocol, then at protocol completion he has i_Q or the other party Q does not have i_P.

Termination for P: If P behaves according to the protocol, he finishes the protocol after a finite amount of time. After that no changes to the state of P will happen (which means that neither the fairness for this party will be lost nor that a successful exchange will be changed to an aborted exchange or vice versa).

The effectiveness property only rules out trivial protocols that do not exchange anything. The fairness property is sometimes called strong fairness [ASW97, Aso98] or atomicity [Tyg96] and ensures that party P will not be disadvantaged in the exchange, even if Q misbehaves and tries to cheat P. Fair

exchange protocols provide fairness to both parties. The termination property ensures that P can eventually stop the protocol execution, even if Q does not follow the protocol. Note that we use a stronger notion than that of Asokan [Aso98, ASW98] which even allows changes after termination, as long as fairness is not reduced. This would allow the TTP to modify the outcome of the exchange and e.g. finish an aborted exchange by delivering both parties the expected items. Our definition of termination is a little bit stronger to prevent such surprises.

2.3 Item Properties

An exchanged *digital item* is modeled as a bit string which can be transferred to or stored by other parties. These items are assumed to be idempotent, i.e. receiving these items multiple times has the same effect as receiving them once.

Some special item properties — namely generatability and revocability proposed in [ASW97] — can be exploited to implement optimistic protocols. In order to realize our new protocol we have to distinguish a weak and a strong notion of these properties.

Generatability. If an item is generatable, then the TTP has the ability to create this item on its own. We distinguish two notions of generatability:

Strong generatability: The TTP can generate this kind of item.
Weak generatability: The TTP can try to generate this kind of item, but the TTP may fail. In this case the TTP can always determine, which party misbehaved and thereby prevented generating the item.

The implementation of generatability in general follows this pattern: First the owner P of a generatable item i_P has to convince the receiving party Q that i_P is indeed generatable. Therefore, P sends a message including a *generate information* $geninfo_{i_P}$ to Q. Then the TTP will be able to use $geninfo_{i_P}$ to generate i_P. In detail, the generate information $geninfo_{i_P}$ must have the following properties:

- The receiving party Q can verify this information to determine the generatability of i_P. If $geninfo_{i_P}$ is valid, the weak/strong generatability of i_P is proven. If $geninfo_{i_P}$ is invalid, it will be rejected by Q.
- The receiving party Q can send $geninfo_{i_P}$ to the TTP in order to prove that he has the permission to receive the generated i_P in exchange for i_Q. Furthermore, the TTP may utilize $geninfo_{i_P}$ for generating i_P. But then it is important that Q alone cannot derive useful information about i_P from $geninfo_{i_P}$.
- By verifying $geninfo_{i_P}$ the TTP can determine a cheating party.
  - In case of strong generatability a correct $geninfo_{i_P}$ implies that the TTP is always able to generate i_P. An invalid $geninfo_{i_P}$ is rejected as misbehavior of Q.

- In case of weak generatability a correct $geninfo_{i_P}$ is considered as a commitment of P that the TTP will be able to generate i_P. If $geninfo_{i_P}$ is valid and the TTP fails to generate i_P, then the TTP detected the cheating of P who did not fulfill his commitment. An invalid $geninfo_{i_P}$ is always rejected as misbehavior of Q.

We now illustrate these definitions with some examples:

Any digital item can be made strongly generatable by the following method: A party P sends its item i_P with a corresponding description $desc_{i_P}$ to the TTP, which only proceeds, if i_P matches the given description. Then the TTP stores i_P under its description and returns P a signature on $desc_{i_P}$. This value $sig_{TTP}(desc_{i_P})$ serves as the generate information for i_P. By checking this signature a receiving party Q can verify that i_P is strongly generatable. Q only needs to send $sig_{TTP}(desc_{i_P})$ together with the description to the TTP, which will answer this correct generate information with the stored item i_P.

The described method for strong generatability has the disadvantage that the party P must contact the TTP for every item. As this causes much work at the TTP, it is desirable to achieve generatability without this costly interaction with the TTP. One way to realize this is called *verifiable escrow* [Mao97,ASW00] and applies to digital signatures. The generate information $geninfo_{i_P}$ consists of some escrow information for the signature i_P and a proof that this escrow information can indeed be used to generate i_P. This escrow information is in general implemented by a pre-image of the signature i_P which can be converted to a valid signature only by the TTP (see e.g. [ASW00,BF98,Che98,Ate99,MS01] for details). However, the proof for the correctness of the pre-image may cause much computation, especially if it is based on the cut-and-choose paradigm like in [ASW00].

In contrast it is quite simple to achieve weak generatability given an arbitrary digital item. The owner P encrypts the item i_P with the public key of the TTP and signs this encryption $E_{TTP}(i_P)$ and the description of the item. This means that the generate information $geninfo_{i_P}$ consists of $sig_P(E_{TTP}(i_P), desc_{i_P})$ together with the signed values. The generate information is correct, if and only if P signed some ciphertext and the description $desc_{i_P}$. With this signature the party P states that the ciphertext can be decrypted by the TTP and contains an item matching the description $desc_{i_P}$. If the TTP fails to decrypt this ciphertext or the resulting plaintext does not match the description $desc_{i_P}$, then the TTP knows that the signer P must have cheated. Thus, this method ensures weak generatability for any item i_P.

Revocability. If an item is revocable, the TTP has the ability to invalidate this item. We distinguish two notions of revocability:

Strong revocability: The TTP can revoke this kind of item, which makes the item useless for its receiver.

Weak revocability: The TTP can try to revoke this kind of item, but it may fail. In this case the TTP can always be sure, that the receiving party got the item or can still get it.

The implementation of revocability can be more efficient than that of generatability. As the sender P of a revocable item may later depend on this property, it is his own responsibility that revocation is possible. Therefore, it is not required to contact Q in advance to ensure revocability.

The following examples illustrate the definitions of strong and weak revocability: If an electronic payment system supports cancellation of payments on behalf of the TTP, then payments are strongly revocable. If cancellation of these electronic payments can fail due to the behavior of the receiver, only weak revocability can be achieved. For example, the receiver of the payment might be able to withdraw all his money and close his bank account. Then the bank cannot revoke the payment, but it can tell the TTP that the receiver got the payment, which ensures weak revocability.

3 Previous Optimistic Fair Exchange Protocols and Their Problems

Fair exchange of revocable items has received only little attention in the literature, as revocation is usually only possible in payment systems. But especially for anonymous electronic payment systems [Cha83] it is important to use fair exchange protocols, as an anonymous party is more likely to exploit an unfair exchange. Also the opposite may be a problem: Without fair exchange a merchant may try to cheat anonymous customers, as these would need to give up their anonymity, if they e.g. want to complain at the bank to get their money back. Thus, we think that optimistic fair exchange of electronic payments is a critical issue and deserves a thorough investigation.

A previously proposed fair exchange solution is the payment for receipt protocol of [ASW98, Section 5.2], which however fails to provide either fairness or termination to the payee. The exchange protocol for payments vs. digital goods of [PV99,VPG99] fails to guarantee termination for the payee. The protocols for revocable items in [ASW97] and [Sch00] only work in the synchronous system model, which has the drawbacks discussed in [ASW00, Section 2.2].

The deficiencies of all these protocols for revocable items led us to the question whether it is possible to construct an asynchronous optimistic fair exchange protocol, which exploits revocability. Our protocol solving this question is presented in Section 4.

An alternative approach for asynchronous optimistic fair exchange is only based on the generatability property, and many of these protocols for generatable items are discussed in the literature (e.g. [GJM99,ZDB99,ASW00,MS01,MK01]). But all of these protocols are based on the same protocol pattern introduced by Asokan [Aso98,ASW98]. This protocol requires one strongly generatable and one weakly generatable item.

If Asokan's protocol shall be applied to the exchange of electronic payments, then the precondition of strong generatability has to be fulfilled. One example where it is especially difficult to use Asokan's protocol is the exchange of anonymous electronic coins for arbitrary digital goods. As no efficient method (i.e.

without TTP interaction) is known to make arbitrary goods strongly generatable, we only assume that they are weakly generatable, which can be achieved with the technique described in Section 2.3. This however implies that we need to make the payment strongly generatable to use Asokan's exchange protocol. Thus, we discuss the generatability property based on the example of Chaum's anonymous online coins [Cha83] which are usually called ECash [Sch97]. At payment these coins simply consist of the bank's signature on a randomly chosen serial number m. This signature $sig_{bank}(m)$ can in principle be made strongly generatable using a verifiable escrow technique for this signature scheme. But even if the signature is strongly generatable, the generation of the *coin* can fail, as the coin might have been deposited to a different account in the meantime. This results in no more than weak generatability for the coin, and this is not enough to execute Asokan's protocol which requires strong generatability.

Besides this difficulty of making an anonymous electronic coin strongly generatable, there also exists another problem. In order to check the validity of the generate information for the coin, we must reveal the serial number m, as this is the only way to detect whether this coin has been deposited before. In Asokan's protocol the merchant can abort the exchange after he has received the generate information. This ensures that the merchant can never get the coin from the TTP, but the merchant already got the serial number m contained in the generate information. As this violates the unlinkability property of the payment system, the payer will be forced to refresh these coins by returning them to the bank and retrieving new anonymous coins in return. This linkability problem obviously causes an additional overhead when strong generatability shall be implemented in anonymous electronic payment systems. Even if linkability is acceptable to the payer, such a refresh is still required, if the coins contain double-spending detection mechanisms as proposed for off-line payment systems (e.g. [Bra93, CMS96]).

Our solution to these two problems with strongly generatable coins is to simply make them strongly revocable instead, which is relatively easy to achieve in the case of ECash-like payment systems. The TTP simply asks the bank to cancel a deposit and make some already spent coins valid again. This is already enough to revoke electronic coins. To ensure unlinkability of the revoked coins, we need to refresh them by exchanging them against new ones at the bank. As this refreshment would also be required for generatable coins, this cannot be considered as an overhead.

To summarize this discussion we stress that revocability of electronic payments may be much easier to implement than generatability (especially for payment systems like [Sch97, KV01]). Thus, the exchange protocol which we present in the following section is ideal for the exchange of electronic payment vs. arbitrary goods, as it only requires strongly revocable payments and weakly generatable goods.

4 Fair Exchange Exploiting Strong Revocability

Our new protocol requires a strongly revocable i_A and a weakly generatable i_B. Due to the definition of generatability a strongly generatable i_B would also be correct, as strong generatability implies weak generatability. If these item properties are fulfilled, our protocol ensures effectiveness as well as fairness and termination for both parties. The protocol is optimistic, as the TTP is only required to resolve conflicts, and it works in the asynchronous system model, as it does not rely on bounded message delays or time-outs to ensure fairness.

4.1 Protocol Description

We assume that the parties have already agreed on the descriptions of the exchanged items and the TTP to be used in case of a conflict. Our protocol consists of three parts: The exchange protocol, the resolve sub-protocol, and the abort sub-protocol. In the faultless case only the exchange protocol has to be executed, whereas the sub-protocols are used for conflict resolution by A and B, respectively.

Exchange protocol. The steps of our fair exchange protocol are given in Table 1. Party B starts by sending the generate information for his item. A checks the validity of $geninfo_{i_B}$ and only proceeds by transmitting i_A, if the generate information is valid. Otherwise, A is still able to quit the exchange, as he did not reveal any useful information yet. If A does not receive $geninfo_{i_B}$ in time, he can also resolve this conflict by simply quitting.

B then checks the received item i_A and will start the abort sub-protocol, if this item is invalid. B will start the same action for conflict resolution, if the item i_A has not yet arrived and B does not want to wait for it any longer.

Finally, B answers a correct i_A with his own item i_B. The receiving party A then tests the validity of the received item. If it is valid, the exchange is successfully finished. Otherwise A starts a conflict resolution with the resolve sub-protocol. This sub-protocol must also be executed, if A does not receive i_B.

Table 1. Fair exchange protocol for a strongly revocable item i_A and a weakly generatable item i_B.

B $\rightarrow$ A :	$geninfo_{i_B}$
A	: Verify $geninfo_{i_B}$ (if not OK, quit exchange)
A $\rightarrow$ B :	i_A
B	: Test, whether $check(i_A, desc_{i_A})$ is *true* (if not, start abort sub-protocol)
B $\rightarrow$ A :	i_B
A	: Test, whether $check(i_B, desc_{i_B})$ is *true* (if not, start resolve sub-protocol)

As the communication between A and B is unreliable, one party would be waiting forever, if a message was never sent or got lost during transmission. A

party that is expecting a message and does not want to wait any longer can unilaterally start the conflict resolution procedure to continue the protocol. If one of the messages does not arrive in time, the receiving party simply proceeds as if the expected message were invalid. Thus, the protocol can be continued even without the other party.

Resolve sub-protocol. Party A can resolve a conflict with the protocol given in Table 2. The precondition for this protocol is that A received a valid $geninfo_{i_B}$ from B. Otherwise, A would still be able to simply quit the protocol. First A sends his item, the generate information for i_B, and the descriptions of both items to the TTP, which only processes requests containing a valid item i_A and a correct generate information $geninfo_{i_B}$. If the exchange has already been aborted, the TTP revokes i_A to ensure fairness for A. Otherwise, the TTP tries to finish the exchange by generating i_B. After successfully creating i_B the TTP stores i_A (only if i_A is not already stored) for further requests by B and delivers i_B to A. If the TTP fails in generating i_B, then B must be responsible for this due to the weak generatability of i_B. As $geninfo_{i_B}$ is valid, B did not fulfill his commitment that i_B will be generatable with $geninfo_{i_B}$. Thus, the TTP aborts the exchange by revoking i_A, which ensures fairness for party A.

Table 2. The owner of the revokable item can try to resolve the exchange with this sub-protocol.

```
A  → TTP : i_A, geninfo_{i_B}, desc_{i_A}, desc_{i_B}
  TTP     : If A has sent a valid i_A and geninfo_{i_B}:
                If the exchange has not been aborted yet:
                   If the TTP can generate a valid i_B:
                         TTP     : store i_A
                      TTP →  A : i_B
                   Else:  // generating i_B failed
                         TTP     : revoke i_A
                      TTP →  A : exchange has been aborted
                Else:  // exchange aborted
                         TTP     : revoke i_A
                      TTP →  A : exchange has been aborted
             Else:  // invalid i_A or geninfo_{i_B}:
                TTP →  A : Error: invalid request
```

Abort sub-protocol. If B does not receive i_A during the exchange, he asks the TTP to abort the exchange with the sub-protocol in Table 3. If the TTP has not yet sent i_B to A, it simply aborts the exchange. This ensures that the TTP will never generate i_B for party A.

Table 3. The owner of the generatable item can try to abort the exchange with this sub-protocol.

B $\rightarrow$ TTP : abort request	
TTP	: If the TTP has not yet sent i_B to A:
	TTP $\rightarrow$ B : exchange has been aborted
	Else: // i_B was sent to A
	TTP $\rightarrow$ B : i_A

If A has already resolved the conflict and received the generated i_B, then an abort will not be possible anymore. Instead the TTP forwards the stored item i_A to B which results in a successful exchange.

4.2 Fairness Properties

We now show that the new exchange protocol ensures effectiveness, fairness, and termination for A and B. Thus, our protocol enjoys the same strong fairness guarantees as Asokan's protocol with two generatable items [Aso98, ASW98].

Effectiveness: It is obvious that both parties gain the expected items, if the exchange protocol is executed without errors.

Fairness: We show the fairness property for both parties and start with party A. Fairness for A means that it will never happen that B gains i_A while A does not receive i_B. Therefore we analyze the two possibilities how B can receive i_A and show that in this case A always obtains i_B:

1. If B gains i_A during the exchange protocol, then A has received a valid $geninfo_{i_B}$ before. A may then get i_B directly from B or invoke the resolve sub-protocol. If the TTP can generate i_B, fairness for A is established. If the TTP fails to generate i_B in spite of a correct $geninfo_{i_B}$, then i_A will be revoked, which makes this item useless for B and ensures fairness for A.
2. If B gains i_A as an answer in the abort protocol, then A has received i_B in the resolve sub-protocol before.

This proves the fairness property for A. For the fairness of B we have to analyze the two cases in which A receives i_B. Then an honest B must always be able to get i_A.

1. If A gains i_B during the exchange protocol, then B has received i_A before. An honest B can then be sure that the item i_A will never be revoked, as revocation is only performed after B prevented generatability of i_B or started the abort protocol in spite of any failure. In both cases B misbehaved so that we don't need to ensure fairness for him.
2. If A gains i_B in the resolve protocol, then the TTP has stored i_A. If B has not yet received i_B, he can still receive it with the abort protocol at any time.

Termination: We first show that both parties can terminate in finite time. Then we show that they do not have to fear any state changes after termination.

A and B have the following possibilities to eventually finish the exchange:

1. A finishes the exchange after he has received a correct i_B from B.
2. B can finish the exchange after he has received a correct i_A from A.
3. If A does not receive a correct $geninfo_{i_B}$, he simply quits the protocol without exchanging anything.
4. If A does not receive a correct i_B after sending i_A to B, he starts the resolve sub-protocol, which finishes the exchange either successfully or by aborting. Only a misbehaving A may fail to terminate in the resolve sub-protocol, but this is irrelevant for the termination property, which only has to be ensured for honest parties.
5. If B does not receive a correct i_A after sending $geninfo_{i_B}$ to A, he starts the abort sub-protocol, which either aborts the exchange or finishes it successfully.

After termination no state changes which degrade fairness can occur, as we have already proven the fairness property for both parties. Thus, we only have to check that a successful or aborted termination state will never change for honest parties. It is sufficient to look at the conflict resolution protocols, as only the TTP can change the state of a terminated party by generating or revoking items.

- If i_B is generated, then B has not terminated in the state "aborted", as this would require the successful execution of the abort sub-protocol, which would prevent the generation of i_B. Thus, if B has terminated, he must have received the item i_A, which implies a "success" state at termination.
- If i_A is revoked, then an honest B has not terminated in the "success" state, as he either executed the abort sub-protocol or misbehaved by preventing the generation of i_B. Thus, an honest B must have terminated in the "aborted" state.

In both cases the terminated party will never change its state, which proves the termination property for both parties.

4.3 Discussion

Our new asynchronous optimistic fair exchange protocol is the first that exploits revocable items and does not require strongly generatable items. Our protocol offers the same fairness guarantees as Asokan's protocol with two generatable items. Due to its simple structure our protocol has some advantages:

- Our protocol is very efficient, as it requires only 3 messages in the faultless case.
- Conflict resolution is quite simple: There is the resolve sub-protocol for A and the abort sub-protocol for B.

In contrast, Asokan's protocol needs 4 messages in the faultless case, which has been shown to be optimal in the general case of asynchronous optimistic fair exchange without revocability [PSW98]. Furthermore, Asokan's protocol has two resolve protocols and one abort protocol, which makes its implementation rather complicated. This is probably the reason why [ASW98] contains some minor errors, which are discussed in [ZDB00, BK00].

If strong generatability and strong revocability are both available, our new protocol should be chosen due to its advantages over Asokan's protocol.

Our new protocol enhances the applicability of optimistic fair exchange as shown in Figure 1. If one of the exchanged items is strongly revocable, our protocol ensures fair exchange, as we can simply make the other item weakly generatable by the technique described in Section 2.3. If only strong generatability is available, Asokan's protocol should be used for fair exchange. If neither strong revocability nor strong generatability are given, no satisfactory optimistic fair exchange protocols are known. As a last resort we then can rely on the less efficient active protocols (e.g. [BP90, Tyg96, CHTY96, ZG96, FR97]), which need the active participation of the TTP in every exchange transaction.

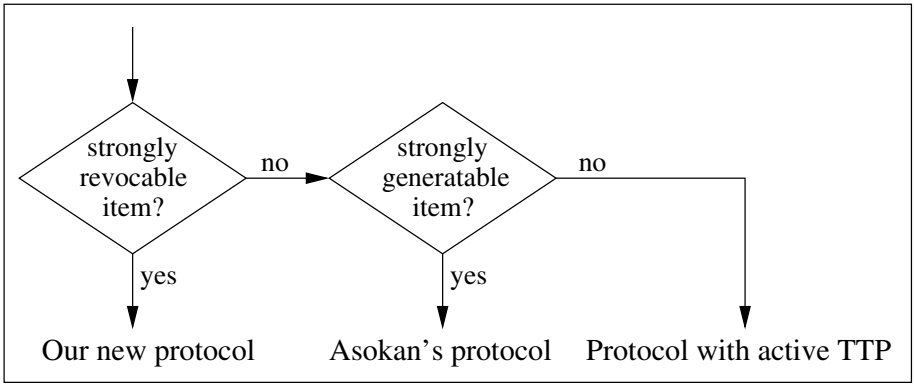

Fig. 1. How to choose an asynchronous fair exchange protocol depending on the item properties.

5 Conclusion

We have presented the first asynchronous optimistic fair exchange protocol that only exploits strong revocability and is not dependant on strong generatability. The additional assumption of our protocol that the non-revocable item has to be weakly generatable can easily be fulfilled by any digital item, as we have shown by the example for weak generatability in Section 2.3.

Compared to the fair exchange protocol of Asokan, which needs a strongly generatable item, our protocol is more efficient and easier to implement: We need less messages in the faultless case and our conflict resolution is less complicated, as we only have to implement two sub-protocols instead of three.

Furthermore, we have shown that the generatability of electronic payments like ECash is difficult to implement and, in contrast to that, revocability may be much easier to realize. Thus, we think that our new protocol is the best currently known protocol for fair exchange of anonymous electronic payments for digital goods.

Acknowledgments. The author would like to thank Felix Gärtner and Henning Pagnia for their feedback and helpful discussions. Also many thanks to the anonymous referees for their detailed comments.

References

[Aso98] N. Asokan. *Fairness in electronic commerce*. PhD thesis, University of Waterloo, Canada, May 1998.

[ASW97] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In Tsutomu Matsumoto, editor, *4th ACM Conference on Computer and Communications Security*, pages 6–17, Zürich, Switzerland, April 1997. ACM Press.

[ASW98] N. Asokan, Victor Shoup, and Michael Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, Oakland, CA, May 1998. IEEE Computer Society Press.

[ASW00] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610, April 2000.

[Ate99] Giuseppe Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proceedings of 6th ACM Conference on Computer and Communications Security (CCS '99)*, pages 138–146, Singapore, November 1999. ACM Press.

[BF98] Colin Boyd and Ernest Foo. Off-line fair payment protocol using convertible signatures. In *Advances in Cryptology – ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 271–285, Beijing, China, October 1998. Springer-Verlag.

[BK00] Colin Boyd and Peter Kearney. Exploring fair exchange protocols using specification animation. In *Information Security – ISW 2000*, volume 1975 of *Lecture Notes in Computer Science*, pages 209–223, Wollongong, Australia, December 2000. Springer-Verlag.

[BN00] Dan Boneh and Moni Naor. Timed commitments. In *Advances in Cryptology – CRYPTO '2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 236–254, Santa Barbara, CA, 2000. Springer-Verlag.

[BOGMR90] Michael Ben-Or, Oded Goldreich, Silvio Micali, and Ronald L. Rivest. A fair protocol for signing contracts. *ACM Transactions on Information Theory*, 36(1):40–46, January 1990.

[BP90] Holger Bürk and Andreas Pfitzmann. Value exchange systems enabling security and unobservability. *Computers & Security*, 9(8):715–721, 1990.

[Bra93] Stefan Brands. Untraceable off-line cash in wallets with observers. In *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318, Santa Barbara, CA, August 1993. Springer-Verlag.

[Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology – CRYPTO '82*, pages 199–203. Plenum, 1983.

[Che98] Liqun Chen. Efficient fair exchange with verifiable confirmation of signatures. In K. Ohta and D. Pei, editors, *Advances in Cryptology – ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 286–299, Beijing, China, 18–22 October 1998. Springer-Verlag.

[CHTY96] Jean Camp, Michael Harkavy, J. D. Tygar, and Bennet Yee. Anonymous atomic transactions. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pages 123–133, Oakland, CA, November 1996.

[CMS96] Jan Camenisch, Ueli Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. In *Computer Security – ESORICS '96*, volume 1146 of *Lecture Notes in Computer Science*, pages 31–43, Rome, Italy, September 1996. Springer-Verlag.

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In *Advances in Cryptology – CRYPTO '82*, pages 205–210, New York, USA, 1982. Plenum Publishing.

[EY80] Shimon Even and Yacov Yacobi. Relations amoung public key signature systems. Technical Report 175, Computer Science Department, Technicon, Haifa, Israel, 1980.

[FR97] Matthew K. Franklin and Michael K. Reiter. Fair exchange with a semi-trusted third party. In Tsutomu Matsumoto, editor, *4th ACM Conference on Computer and Communications Security*, pages 1–5, Zürich, Switzerland, April 1997. ACM Press.

[GJM99] Juan A. Garay, Markus Jakobsson, and Philip MacKenzie. Abuse-free optimistic contract signing. In Michael Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 449–466, Santa Barbara, CA, 15–19 August 1999. Springer-Verlag.

[Jak95] Markus Jakobsson. Ripping coins for fair exchange. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 220–230, St. Malo, France, 21–25 May 1995. Springer-Verlag.

[KV01] Dennis Kügler and Holger Vogt. Marking: A privacy protecting approach against blackmailing. In *Public Key Cryptography – PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 137–152, Cheju Island, Korea, February 2001. Springer-Verlag.

[Mao97] Wenbo Mao. Verifiable escrowed signature. In *Information Security and Privacy – ACISP '97*, volume 1270 of *Lecture Notes in Computer Science*, pages 240–248, Sydney, Australia, July 1997. Springer-Verlag.

[MK01] Olivier Markowitch and Steve Kremer. An optimistic non-repudiation protocol with transparent trusted third party. In *Information Security – ISC 2001*, volume 2200 of *Lecture Notes in Computer Science*, pages 363–378, Malaga, Spain, October 2001. Springer-Verlag.

[MS01] Oliver Markowitch and Shahrokh Saeednia. Optimistic fair exchange with transparent signature recovery. In *Financial Cryptography – FC 2001*, volume 2339 of *Lecture Notes in Computer Science*, pages 339–350, Grand Cayman, British West Indies, 19–22 February 2001. Springer-Verlag.

[PSW98] Birgit Pfitzmann, Matthias Schunter, and Michael Waidner. Optimal efficiency of optimistic contract signing. In *Proceedings of the 17th Symposium on Principles of Distributed Computing (PODC '98)*, pages 113–122, New York, 1998. ACM Press.

[PV99] Henning Pagnia and Holger Vogt. Exchanging goods and payment in electronic business transactions. In *Proceedings of the Third European Research Seminar on Advances in Distributed Systems (ERSADS)*, Madeira Island, Portugal, April 1999.

[Sch97] Berry Schoenmakers. Security aspects of the ecash payment system. In *COSIC '97 Course*, volume 1528 of *Lecture Notes in Computer Science*, pages 338–352, Leuven, Belgium, June 1997. Springer-Verlag.

[Sch00] Matthias Schunter. *Optimistic Fair Exchange*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, October 2000.

[Tyg96] J. D. Tygar. Atomicity in electronic commerce. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC '96)*, pages 8–26, Philadelphia, PA, May 1996. ACM Press.

[VPG99] Holger Vogt, Henning Pagnia, and Felix C. Gärtner. Modular fair exchange protocols for electronic commerce. In *Proceedings of the 15th Annual Computer Security Applications Conference*, pages 3–11, Phoenix, Arizona, December 1999. IEEE Computer Society Press.

[XYZZ99] Shouhuai Xu, Moti Yung, Gendu Zhang, and Hong Zhu. Money conservation via atomicity in fair off-line e-cash. In *Information Security – ISW'99*, volume 1729 of *Lecture Notes in Computer Science*, pages 14–31, Kuala Lumpur, Malaysia, November 1999. Springer-Verlag.

[ZDB99] Jianying Zhou, Robert Deng, and Feng Bao. Evolution of fair non-repudiation with TTP. In *Information Security and Privacy – ACISP '99*, volume 1587 of *Lecture Notes in Computer Science*, pages 258–269, Wollongong, Australia, 7–9 April 1999. Springer-Verlag.

[ZDB00] Jianying Zhou, Robert Deng, and Feng Bao. Some remarks on a fair exchange protocol. In *Public Key Cryptography – PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 46–57, Melbourne, Australia, January 2000. Springer-Verlag.

[ZG96] Jianying Zhou and Dieter Gollmann. A fair non-repudiation protocol. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 55–61, Oakland, CA, May 1996. IEEE Computer Society Press.

Fully Private Auctions in a Constant Number of Rounds

Felix Brandt

Computer Science Department
Technical University of Munich
brandtf@cs.tum.edu

Abstract. We present a new cryptographic auction protocol that prevents extraction of bid information despite any collusion of participants. This requirement is stronger than common assumptions in existing protocols that prohibit the collusion of certain third-parties (e.g. distinct auctioneers). Full privacy is obtained by using homomorphic ElGamal encryption and a private key that is distributed among the set of bidders. Bidders jointly compute the auction outcome on their own without uncovering any additional information in a constant number of rounds (three in the random oracle model). No auctioneers or other trusted third parties are needed to resolve the auction. Yet, robustness is assured due to public verifiability of the entire protocol. The scheme can be applied to any uniform-price (or so-called $(M+1)$st-price) auction. An additional, optional, feature of the protocol is that the selling price is only revealed to the seller and the winning bidders themselves. We furthermore provide an in-depth analysis of ties in our protocol and sketch a scheme that requires more rounds but is computationally much more efficient.

1 Introduction

Auctions have become the major phenomenon of electronic commerce during the last years. In recent times, the need for privacy has been a factor of increasing importance in auction design and various schemes to ensure the safe conduction of sealed-bid auctions have been proposed.

We consider a situation where one seller and n bidders or buyers intend to come to an agreement on the selling of a good[1]. Each bidder submits a sealed bid expressing how much he is willing to pay. The bidders want the highest bidder to win the auction for a price that has to be determined by a publicly known rule (e.g. the highest or second-highest bid). In order to fulfill this task, they need a trusted third-party, which is called the "auctioneer". Among the different auction protocols, the second-price or so-called Vickrey auction [1], where the highest bidder wins by paying the amount of the second-highest bid, has received particular attention in recent times because it is "strategy-proof", i.e., bidders are always best off bidding their private valuation of a good. This

[1] The assignment of tasks in reverse auctions works similarly.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 223–238, 2003.
© Springer-Verlag Berlin Heidelberg 2003

is a huge advantage over first-price auctions, where bidders have to estimate the other bidders' valuations when calculating their bid. However, despite its impressive theoretical properties, the Vickrey auction is rarely used in practice. It is generally agreed [2,3,4] that the Vickrey auction's sparseness is due to two major reasons: the *fear of an untruthful auctioneer* and the *reluctance of bidders to reveal their true valuations*. The winner of an auction has to doubt whether the price the auctioneer tells him to pay is actually the second-highest bid. The auctioneer could easily make up a "second-highest" bid to increase his (or the seller's) revenue. In addition to a possibly insincere auctioneer, bidders have to reveal their valuations to the auctioneer. There are numerous ways to misuse these values by giving them away to other bidders or the seller [5,6,7]. It remains in the hands of the auctioneer whether the auction really is a *sealed*-bid auction.

The proposed protocol removes both crucial weaknesses of the Vickrey auction by omitting the auctioneer and distributing the calculation of the selling price on the bidders themselves. No information concerning the bids is revealed unless *all* bidders share their knowledge, which obviously uncovers all bids in any auction protocol. Furthermore, our protocol is applicable to a generalization of the Vickrey auction called uniform-price or $(M+1)$st-price auction. In an $(M+1)$st-price auction, the seller offers M identical items and each bidder intends to buy *one* of them. It has been proven that it is an strategy-proof mechanism to sell those items to the M highest bidders for the uniform price given by the $(M+1)$st highest bid [1,8]. The Vickrey auction is just a special case of this mechanism for the selling of single goods ($M = 1$).

Thus, our main contribution is a verifiable protocol for n participants, each having a secret value, that only reveals the $(M+1)$st highest value to the M participants who possess higher values.

The remainder of this paper is structured as follows. Section 2 summarizes existing efforts in the field of cryptographic auction protocols. Section 3 defines essential attributes that ensure a secure and private auction conduction and Section 4 introduces "bidder-resolved auctions". In Section 5, we propose a bidder-resolved $(M+1)$st-price auction protocol. The paper concludes with an overview of the protocol's complexity and a brief outlook in Section 6.

2 Related Work

There has been a very fast-growing interest in cryptographic protocols for auctions during the last years. In particular, Vickrey auctions and recently the more general $(M+1)$st-price auctions attracted much attention. Starting with the work by Franklin and Reiter [9], which introduced the basic problems of sealed-bid auctions, but disregarded the privacy of bids after the auction is finished, many secure auction mechanisms have been proposed, e.g. [7,10,11,12,13,14,15,16,17, 18,19,20,21,22,23,24,25,26].

When taking away all the protocols that (in their current form) are only suitable for the secure execution of *first*-price auctions or reveal (partial) infor-

mation after the auction is finished [7,9,11,15,19,22,23,27,25,26], the remaining work can be divided into two categories.

Most of the publications rely on threshold computation that is distributed among auctioneers [14,16,17,18,24]. This technique requires m auctioneers, out of which a fraction (mostly a majority) must be trustworthy. Bidders send shares of their bids to each auctioneer. The auctioneers jointly compute the selling price without ever knowing a single bid. This is achieved by using techniques like verifiable secret sharing and secure multiparty function evaluation. However, a collusion of, e.g., three out of five auctioneer servers can already exploit the bidders' trust. We argue that distributing the trust onto several distinct auctioneers does not solve the privacy problem, because you can never rule out that some of them, or even *all* of them, collude.

The remaining auction protocols prune the auctioneer's ability to falsify the auction outcome and reveal confidential information by introducing a new third-party that is not *fully* trusted. However, all of these approaches make weak assumptions about the trustworthiness of this third-party. In [12,13] the third-party may not collude with any participating bidder; in [20,21] it is prohibited that the third-party and the auctioneer collude. A recent scheme [10] uses a homomorphic, indistinguishable public-key encryption scheme like ElGamal to compute on encrypted bids. However, the private key is either held by a trusted third-party or is shared among a set of confidants which makes the protocol as safe as the ones using several auctioneers (see Section 4 for information on how this scheme can be distributed on bidders).

Concluding, all present work on secure auctions more or less relies on the exclusion of third-party collusion, may it be auctioneers or other semi-trusted institutions. Additionally, many of the existing schemes publicly announce the winner's identity and all of them declare the selling price rather than making this information only visible to the seller and the winners.

3 General Assumptions

This section specifies demands that a secure auction protocol has to meet. Furthermore, we make several rigorous assumptions about auction participants and collusions between them. The required properties for safe conductions of sealed-bid auctions can be divided into two categories.

Privacy. *No information concerning bids and the corresponding bidders' identities is revealed during and after the auction.*

The only information that naturally has to be delivered is the information that is needed to carry out the transaction, i.e., the winning bidders and the seller learn the selling price and the seller gets to know the winners' identities. As [27] pointed out, *anonymity* of the winners is crucial. Otherwise, a bidder that breaks a collusive agreement could be identified by his partners. [11] introduced the property of "receipt-freeness" in the context of auctions.

It prevents bidders from proving their bidding prices in order to circumvent bid-rigging. Our protocol is not receipt-free as this would heavily affect efficiency and because receipt-freeness requires untappable channels.

Privacy, as we understand it, implies that no information on any bid is revealed to the public, in particular no bid statistics (e.g. the amount of the lowest bid or an upper bound for the highest bid) can be extracted.

Correctness. *The winner and the selling price are determined correctly.*

This requirement includes *non-repudiation* (winning bidders cannot deny having made winning bids) and the immutability of bids. *Robustness* (no subset of malicious bidders can render the auction outcome invalid) also belongs to this category (see Section 4).

Privacy and correctness have to be ensured in a hostile environment as we allow every feasible type of collusion. We assume that up to $n-1$ bidders might share their knowledge and act as a team. This implies that each bidder can have arbitrarily many bidder sub-agents, controlled by him. Besides, the seller might collude with bidders, and any number of auctioneers or other third parties might collude and are therefore not trustworthy. We assume the standard model of a secure broadcast channel.

4 Bidder-Resolved Auctions

According to the assumptions of the previous section, bidders cannot trust any third-party. We therefore distribute the trust onto the bidders themselves. This allows us to set a new standard for privacy. In a scenario with m auctioneers it cannot be ruled out that all of them collude. However, when distributing the computation on n bidders, we can assume that *all* bidders will never share their knowledge due to the competition between them. If they did, each of them would completely abandon his own privacy, resulting in a public auction. We therefore argue, that only bidder-resolved auctions provide *full privacy*, i.e., no information on any bid can be retrieved unless all bidders collude. Full privacy can be interpreted as $(n-1)$-privacy or (n,n)-threshold privacy.

It is difficult to assure robustness in bidder-resolved auctions. However, verifiability can be used to provide what we call *weak robustness*, so that malicious bidders will be detected immediately (without additional communication and information revelation) and can be excluded from the set of bidders. The protocol can then be restarted with the remaining bidders proving that their bids did not change[2]. This guarantees termination (after at most $n-M$ iterations) and correctness (if we agree that the removal of malicious bidders (and their bids) does not violate correctness). As malicious bidders can easily be fined and they do not gain any information, there should be no incentive to perturb the auction and we henceforth assume that a single protocol run suffices.

[2] This is not mandatory as their should be no reason to strategically change a bid after a bidder has been excluded (assuming the private-value model).

Public verifiability of the protocol is sufficient to provide weak robustness and verifiability can be easily achieved by using zero-knowledge proofs. Unfortunately, when abandoning (strong) robustness, we also lose "fairness". Typically, in the end of a protocol run, each participant holds a share of the result. As simultaneous publication of these shares is impossible, a malicious agent might quit the protocol after having learned the result but before others were able to learn it. There are various techniques to approximate fairness by gradually releasing parts of the secrets to be swapped. Another possibility is to introduce a third-party that publishes the outcome after it received all shares. This third-party does not learn confidential information. It is only assumed not to leave the protocol prematurely. We believe that in auctions with a single seller, it is practical to assign this role to the seller. This obviously leaves the possibility of a "cheating seller" who quits the protocol after having learned the (possibly unsatisfying) result. However, such a seller could be forced to sell the good for the resulting price as bidders can compute the auction outcome on their own (or with another fairness-providing third party).

The naive approach to build a Boolean circuit that computes the auction outcome on binary representations of bids by applying a general multiparty computation (MPC) scheme is not feasible as those schemes are quite inefficient and the circuit depth, and thus the round complexity, depends on the number of bidders and the bid size. Like many other existing schemes, we therefore use an ordered set of k possible prices (or valuations) $(p_1, p_2, \ldots, p_k)$. This results in linear computational complexity but enables special purpose protocols that do not require a general MPC scheme. In fact, our protocol has constant round complexity because it only uses additions and no multiplications.

A framework for bidder-resolved auction protocols could look like this:

- The seller publicly announces the selling of a certain good by publishing
  - the good's description,
  - the amount of units to be sold,
  - the registration deadline,
  - lower and upper bounds of the valuation interval, and
  - a function that prescribes how and how many valuations $(p_1, p_2, \ldots, p_k)$ are distributed among that interval subject to the number of bidders n (enabling linear, logarithmic, or any other form of scaling)

  on a blackboard.
- Interested bidders publish their id's on the blackboard.

  — *registration deadline* —
- The bidders jointly compute the winners and the selling price.

A threshold-scheme providing t-resilience is not appropriate when information is shared among bidders, as any group of bidders might collude due to the assumptions of the previous section. As a consequence, we cannot simply adapt existing auction protocols that were designed for multiple auctioneers. Protocols like [14], or [16] rely on information-theoretic secure multiparty computation according to Ben-Or, Goldwasser and Wigderson [28], which provides at most

insufficient $\lfloor \frac{n-1}{2} \rfloor$-privacy due to the multiplication of degree n polynomials. Another recent protocol by Abe and Suzuki [10] uses verifiable mix and match [15] of ElGamal ciphertexts assuming an honest majority due to robustness requirements. When relaxing these requirements in order to realize a bidder-resolved protocol and discarding binary search to minimize the round complexity, mixing would still require $\mathcal{O}(n)$ rounds. With further changes that enable privacy of the selling price, the computational and message complexity per bidder would be $\mathcal{O}(nkM \log(M))$, which is fairly good as M is negligibly small in many cases (e.g. in a Vickrey auction). However, the number of rounds depends on the number of bidders n.

5 Protocol Description

We will use an additive vector notation to describe our approach. The actual implementation described in Section 5.1, however, will take place in a multiplicative group using ElGamal encryption with a public key that is jointly created by all bidders.
Each bidder sets the bid vector[3]

$$\boldsymbol{b}_i = (b_{i1}, b_{i2}, \ldots, b_{ik}) = (\underbrace{0, \ldots, 0}_{b_i - 1}, Y, \underbrace{0, \ldots, 0}_{k - b_i})$$

according to his bid $b_i \in \{1, 2, \ldots, k\}$, publishes its encryption, and shows its correctness by proving $\forall j \in \{1, 2, \ldots, k\} : \; b_{ij} \in \{0, Y\}$ and $\sum_{j=1}^{k} b_{ij} = Y$ in zero-knowledge manner (like in [10]). $Y \neq 0$ is a generally known group element, e.g. 1.

The homomorphic encryption scheme allows verifiable computation of linear combinations of secrets in a single round. When computing on vectors of homomorphically encrypted values (like $\boldsymbol{b}_i$), this means that besides addition and substraction of (encrypted) vectors, multiplication with (known) matrices is feasible. For example, the "integrated"[10] bid vector

$$\boldsymbol{b}'_i = (\underbrace{Y, \ldots, Y}_{b_i}, \underbrace{0, \ldots, 0}_{k - b_i}) = (b_{i1} + b'_{i2}, b_{i2} + b'_{i3}, \ldots, b_{ik})$$

can be derived by multiplying the bid vector with the $k \times k$ lower triangular matrix L ($\boldsymbol{b}'_i = \mathsf{L}\boldsymbol{b}_i$).

$$\mathsf{L} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \qquad \text{(lower triangular matrix)}$$

[3] To save space, vector components are listed horizontally (bottom-up).

Multiplying a vector with $\mathsf{L}-\mathsf{I}$, where I is the $k \times k$ identity matrix, yields $\boldsymbol{b}'_i$ shifted down by one component.

$$\mathsf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \text{(identity matrix)}$$

If we sum up all integrated bid vectors and down-shifted integrated bid vectors, we obtain a vector that has the following structure (let us for now disregard the possibility of equal bids, we will refer to this case in Section 5.2).

$$(2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i = (\ldots, 6Y, \ldots, 6Y, 5Y, 4Y, \ldots, 4Y, 3Y, 2Y, \ldots, 2Y, Y, 0, \ldots, 0)$$

The position of the (single) component that equals $3Y$ denotes the second-highest bid, $5Y$ the third-highest bid, and so forth. Subtracting $(2M+1)Y\boldsymbol{e}$ with $\boldsymbol{e} = (1, \ldots, 1)$, thus yields a vector in which the component, that refers to the amount of the $(M+1)$st highest bid, is 0. All other components are not 0.
As we intend to create personal indicators for each bidder, we mask the resulting vector so that only winning bidders can read the selling price. This is achieved by adding $\mathsf{U}\boldsymbol{b}_i$.

$$\mathsf{U} = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \text{(upper triangular matrix)}$$

For an arbitrary bidder a, the vector $(2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i - (2M+1)Y\boldsymbol{e} + (2M+2)\mathsf{U}\boldsymbol{b}_a$ only contains a component equal 0, when a qualifies as a winner of the auction. The position of this component then indicates the selling price.
In order to get rid of all information besides the selling price, each component is multiplied with a different random multiplier M_{ij} that is jointly created and unknown to any subset of bidders. Finally, each bidder's personal indicator vector is computed according to the following equation.

$$\boldsymbol{v}_a = \left((2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i - (2M+1)Y\boldsymbol{e} + (2M+2)\mathsf{U}\boldsymbol{b}_a\right)\mathsf{R}^*_a$$

$$\mathsf{R}^*_i = \begin{pmatrix} M_{ik} & 0 & \cdots & 0 \\ 0 & M_{i,k-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & M_{i1} \end{pmatrix} \qquad \text{(random multiplication matrix)}$$

Be aware that R_i^* does *not* represent a feasible linear operation on encrypted values as the homomorphic property only provides addition, but not multiplication, of secrets. The components M_{ij} are unknown to bidders. In Section 5.1, we will present a very efficient way to randomize ElGamal encrypted vector components.

The invariant of the "blinding" transformation are components that equal 0. As described before, those components mark the selling price to winning bidders. Only bidder i and the seller get to know $\boldsymbol{v}_i$.

$$v_{ij} = 0 \quad \Longleftrightarrow \quad \text{Bidder } i \text{ won and has to pay } p_j$$

The following simple example for two bidders illustrates the functionality of the protocol. The computations take place in $\mathbb{Z}_{11}$ and the auction to be conducted is a Vickrey auction ($M = 1$). Bids are $b_1 = 2$ and $b_2 = 5$: $\boldsymbol{b}_1 = (0,1,0,0,0,0)$, $\boldsymbol{b}_2 = (0,0,0,0,1,0)$. The selling price can be determined by computing

$$\begin{pmatrix} 1&0&0&0&0&0\\ 2&1&0&0&0&0\\ 2&2&1&0&0&0\\ 2&2&2&1&0&0\\ 2&2&2&2&1&0\\ 2&2&2&2&2&1 \end{pmatrix} \left(\begin{pmatrix} 0\\0\\0\\0\\1\\0 \end{pmatrix} + \begin{pmatrix} 0\\1\\0\\0\\0\\0 \end{pmatrix} \right) - \begin{pmatrix} 3\\3\\3\\3\\3\\3 \end{pmatrix} = \begin{pmatrix} 0\\1\\2\\2\\3\\4 \end{pmatrix} - \begin{pmatrix} 3\\3\\3\\3\\3\\3 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} .$$

Now, the selling price has to be masked to losing bidders. Bidder 1 is unable to identify the selling price. His indication vector ($\boldsymbol{v}_1$) contains random numbers.

$$\begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + 4 \begin{pmatrix} 1&1&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 0\\0\\0\\0\\1\\0 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + \begin{pmatrix} 4\\4\\4\\4\\4\\0 \end{pmatrix} = \begin{pmatrix} 1\\2\\3\\3\\4\\1 \end{pmatrix} \xrightarrow{\times R_1^*} \begin{pmatrix} .\\.\\.\\.\\.\\. \end{pmatrix}$$

Bidder 2's indicator $\boldsymbol{v}_2$, however, indicates the selling price at the second component (bottom-up).

$$\begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + 4 \begin{pmatrix} 1&1&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 0\\1\\0\\0\\0\\0 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + \begin{pmatrix} 4\\4\\0\\0\\0\\0 \end{pmatrix} = \begin{pmatrix} 1\\2\\10\\10\\0\\1 \end{pmatrix} \xrightarrow{\times R_2^*} \begin{pmatrix} .\\.\\.\\.\\0\\. \end{pmatrix}$$

5.1 Implementation Using ElGamal Encryption

The following implementation of the protocol is based on a distributed version of ElGamal cipher and uses several zero-knowledge proofs like Schnorr's proof of

knowledge of a discrete logarithm [29], Chaum and Pedersen's proof of equality of discrete logarithms [30], and Cramer, Damgård, and Schoenmaker's proof of partial knowledge [31]. It is much more efficient than a previous version [32] that was based on verifiable secret sharing. Indices $+i$ and $\times i$ are used to indicate additive and multiplicative shares, respectively.

ElGamal cipher [33] is a probabilistic public-key cryptosystem that provides two very useful properties: homomorphic and semantically secure encryption [34]. p and q are large primes so that q divides $p-1$. $\mathbb{G}_q$ denotes $\mathbb{Z}_p$'s unique multiplicative subgroup of order q. The private key is $x \in \mathbb{Z}_q$, the public key $y = g^x$ ($g \in \mathbb{G}_q$). A message $m \in \mathbb{G}_q$ is encrypted by computing the ciphertext tuple $(\alpha, \beta) = (my^r, g^r)$ where r is an arbitrary number in $\mathbb{Z}_q$. A message is decrypted by computing $\frac{\alpha}{\beta^x} = \frac{my^a}{(g^a)^x} = m$. The product of two ciphertexts $(\alpha\alpha', \beta\beta')$ represents an encryption of the plaintexts' product mm' (homomorphic property). We will now describe how to apply the ElGamal cryptosystem as a fully private, i.e. non-threshold, multiparty computation scheme.

Distributed key generation: Each participant chooses x_{+i} at random and publishes $y_{\times i} = g^{x_{+i}}$ along with a zero-knowledge proof of knowledge of $y_{\times i}$'s discrete logarithm using [29]. The public key is $y = \prod_{i=1}^{n} y_{\times i}$, the private key is $x = \sum_{i=1}^{n} x_{+i}$. The broadcast round complexity and the computational complexity of the key generation are $\mathcal{O}(1)$.

Distributed decryption: Given an encrypted message (α, β), each participant publishes $\beta_{\times i} = \beta^{x_{+i}}$ and proves its correctness (as described in [30]). The plaintext can be derived by computing $\frac{\alpha}{\prod_{i=1}^{n} \beta_{\times i}}$. Like the key generation, the decryption can be performed in constant time.

Random Exponentiation: A given encrypted value (α, β) can easily be raised to the power of an unknown random number $M = \sum_{i=1}^{n} m_{+i}$ whose addends can be freely chosen by the participants if each bidder publishes $(\alpha^{m_{+i}}, \beta^{m_{+i}})$ and proves the equality of logarithms. The product of the published ciphertexts yields (α^M, β^M). Random Exponentiation can thus be executed simultaneously with distributed decryption in a single step. Random exponentiation has been the bottleneck of our previous auction protocol [32] that was based on verifiable secret sharing.

What follows is the step-by-step protocol specification for bidder a and his bid b_a. $i, h \in \{1, 2, \ldots, n\}$, and $j, b_a \in \{1, 2, \ldots, k\}$. $Y \in \mathbb{G}_q \backslash \{1\}$ is known to all bidders.

1. Choose x_{+a} and $\forall i, j:\ m_{ij}^{+a}, r_{aj} \in \mathbb{Z}_q^*$ at random.
2. Publish $y_{\times a} = g^{x_{+a}}$ along with a zero-knowledge proof of knowledge of $y_{\times a}$'s discrete logarithm using [29]. Compute $y = \prod_{i=1}^{n} y_{\times i}$.
3. $\forall j:$ Set $b_{aj} = \begin{cases} Y & \text{if } j = b_a \\ 1 & \text{else} \end{cases}$ and publish $\alpha_{aj} = b_{aj} y^{r_{aj}}$ and $\beta_{aj} = g^{r_{aj}}$.
4. Prove that $\forall j:\ \alpha_{aj} \in \{Yy^{r_{aj}}, y^{r_{aj}}\}$ ([31]) and $\prod_{j=1}^{k} \alpha_{aj} = Yy^{r_a}$.

5. Compute $\forall i,j : \quad \gamma_{ij} = \dfrac{\prod_{h=1}^{n}\prod_{d=j}^{k}(\alpha_{hd}\alpha_{h,d+1})\left(\prod_{d=1}^{j}\alpha_{id}\right)^{2M+2}}{(2M+1)Y}$ and $\delta_{ij} = \prod_{h=1}^{n}\prod_{d=j}^{k}(\beta_{hd}\beta_{h,d+1})\left(\prod_{d=1}^{j}\beta_{id}\right)^{2M+2}$.
6. Send $\forall i,j : \ \gamma_{ij}^{\times a} = (\gamma_{ij})^{m_{ij}^{+a}}$ and $\delta_{ij}^{\times a} = (\delta_{ij})^{m_{ij}^{+a}x_{+a}}$ with a proof of their correctness ([30]) to the seller who publishes all $\gamma_{ij}^{\times h}$ and $\delta_{ij}^{\times h}$ and the corresponding proofs of correctness for each $i,j,h \neq i$ after having received all of them.
7. Compute $v_{aj} = \dfrac{\prod_{i=1}^{n}\gamma_{aj}^{\times i}}{\prod_{i=1}^{n}\delta_{aj}^{\times i}}$.
8. If $v_{aw} = 1$ for any w, then bidder a is a winner of the auction. p_w is the selling price.

The final steps are conducted in a way that allows the seller to assemble all decrypted indicators before the bidders can compute them. This prevents a winning bidder from aborting the protocol after having learned the auction result. Alternatively, a sub-protocol that enables "fair exchange of secrets" could be used while including the seller into the secret sharing process.
Assuming the random oracle model that allows non-interactive zero-knowledge proofs, the entire protocol requires just three rounds of interaction.

5.2 The Problem of Equal Bids

When two or more bidders have the $(M+1)$st highest bid in common, the protocol yields no winners. There is no information revelation in this case, except that there has been a tie on the $(M+1)$st highest bid. However, this might be used by a group of malicious bidders who submit equal bids on purpose to learn about the selling price. If the tie is undetected, their bids were lower than the selling price. If the protocol fails, their bids were at least as high as the selling price would have been (without their participation). Besides, ties can be used to destroy the protocol's robustness, as tieing bidders can anonymously disrupt the auction. In the following, we will discuss three different methods to circumvent the tie problem. The first two avoid ties while the last one identifies ties.

"Interlacing" Vector Components (Int.) A straight-forward way to avoid the problem is to increase the number of components in $\boldsymbol{v}_i$ from k to nk and insert bidder i's bid in row $nj+i-1$. This increases the computational complexity to $\mathcal{O}(n^2k)$. Unfortunately, this method reveals the identity of one of the $(M+1)$st highest bidders to the winners.

Preventing Equal Bids (Pre). Exact bid amounts b_i can be computed by summing up the components of $\mathsf{L}\boldsymbol{b}_i$. The equality of bids can be detected by computing $(b_i - b_h)M_{ih}$ for each pair of bids, requiring $\frac{n^2-n}{2}$ comparisons. When

equal bids have been detected, k extra rows might be inserted similar to the previous technique. As $n < k$ in most reasonable auction settings, the computational complexity per bidder remains $\mathcal{O}(nk)$ when bids are pairwise different. The exact complexity is $\mathcal{O}(nkT)$, where $T = n - |\{b_i\}_{i=1}^n| + 1$. This technique is generally less complex than the previous one (they are equally complex for the extreme case when all bids are equal). Due to the revelation of equal bids, there is no incentive for malicious bidders to use ties on purpose anymore. However, malicious bidders can try to "guess" bids, i.e., they submit various differing bids and hope for ties, because ties reveal opponents' bids.

Determining Ties (DET). Instead of trying to avoid ties, we can locate the position of ties. As mentioned before, ties only inhibit the protocol when they occur at the $(M+1)$st-highest bid. For this reason, "bad" ties always indicate the selling price. The following method marks ties if they prevent the regular protocol from working. $\sum_{i=1}^n \boldsymbol{b}_i - t\boldsymbol{e}$ is a vector that contains zeros if t bidders share the same bid at the corresponding position $(1 < t \leq n)$. "Good" ties can be masked by adding $(n+1)\left(\mathsf{L}\sum_{i=1}^n \boldsymbol{b}_i - (t+u)\boldsymbol{e}\right)$ where $0 \leq u \leq M$ and $M+1 \leq t+u \leq n$. The resulting vector contains a zero when t bids are equal and there are u bids higher than the tie. The preceding factor $(n+1)$ is large enough to ensure that both addends do not add up to zero. Finally, the position of the tie (which is the selling price) has to be made invisible to losing bidders like in Section 5. This can be done by adding $(n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a$.

Concluding, this method requires the additional computation of indicators

$$\boldsymbol{v}'_{atu} =$$
$$\left(\sum_{i=1}^n \boldsymbol{b}_i - t\boldsymbol{e} + (n+1)\left(\mathsf{L}\sum_{i=1}^n \boldsymbol{b}_i - (t+u)\boldsymbol{e}\right) + (n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a\right)\mathsf{R}^*_{atu} =$$
$$= \left((\mathsf{L}+(n+1)\mathsf{I})\sum_{i=1}^n \boldsymbol{b}_i - (nt+nu+2t+u)\,\boldsymbol{e} + (n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a\right)\mathsf{R}^*_{atu},$$

which increases the overall computational complexity to $\mathcal{O}(n^2kM)$. Information revelation is low compared with the previous two methods if we assume that ties happen "accidently" which can be justified by the fact that there is no gain by using equal bids strategically. Winning bidders learn that the selling price was shared by t bidders and that there were u higher bids. In contrast to the previous two methods, not a single bid origin, i.e. a bidder's identity, is uncovered.

Suppose we have the following compilation of bids ($M = 1$, computation takes place in $\mathbb{Z}_{11}$):

$$\boldsymbol{b}_1 = \begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix}, \quad \boldsymbol{b}_2 = \begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix}, \quad \boldsymbol{b}_3 = \begin{pmatrix}0\\0\\0\\1\\0\\0\end{pmatrix}, \quad \text{and} \quad \boldsymbol{b}_4 = \begin{pmatrix}0\\0\\0\\1\\0\\0\end{pmatrix}.$$

The first two ($t = 2, u \in \{0, 1\}$) indicators look like this (before being masked for each bidder):

$$\begin{pmatrix}0\\2\\0\\2\\0\\0\end{pmatrix} - \begin{pmatrix}2\\2\\2\\2\\2\\2\end{pmatrix} + 5\left(\begin{pmatrix}0\\2\\2\\4\\4\\4\end{pmatrix} - \begin{pmatrix}2\\2\\2\\2\\2\\2\end{pmatrix}\right) = \begin{pmatrix}10\\0\\9\\10\\8\\8\end{pmatrix} \xrightarrow{\times R^*_{1,2,0}} \begin{pmatrix}.\\0\\.\\.\\.\\.\end{pmatrix}$$

$$\begin{pmatrix}0\\2\\0\\2\\0\\0\end{pmatrix} - \begin{pmatrix}2\\2\\2\\2\\2\\2\end{pmatrix} + 5\left(\begin{pmatrix}0\\2\\2\\4\\4\\4\end{pmatrix} - \begin{pmatrix}3\\3\\3\\3\\3\\3\end{pmatrix}\right) = \begin{pmatrix}5\\6\\4\\5\\3\\3\end{pmatrix} \xrightarrow{\times R^*_{1,2,1}} \begin{pmatrix}.\\.\\.\\.\\.\\.\end{pmatrix}$$

For $t > 2$ the first difference contains no zeros, leading to random vectors.

5.3 Round vs. Computational Complexity

General fully private MPC is possible (in the computational model) when assuming weak robustness [35,36]. This means that computational complexity could be drastically reduced by working with the currently most efficient scheme based on homomorphic encryption [37], which allows multiplication of encrypted values in three rounds. Different vectors that indicate prices by zeros can be multiplied into a single vector. This can be used to simplify the computation of indication vectors without any tie problems ($0 \leq u \leq M$).

$$\boldsymbol{v}_{au} = (\mathsf{L} - \mathsf{I}) \sum_{i=1}^{n} \boldsymbol{b}_i - u\boldsymbol{e}, \qquad \boldsymbol{v}_a = \left(\begin{pmatrix}\prod_{u=0}^{M} v_{au1}\\ \prod_{u=0}^{M} v_{au2}\\ \vdots\\ \prod_{u=0}^{M} v_{auj}\end{pmatrix} + \mathsf{U}\boldsymbol{b}_a\right)\mathsf{R}^*_a$$

This technique results in $\mathcal{O}(\log M)$ rounds after all. Additionally, the new structure of $\boldsymbol{v}_a$ enables binary search (in public-price mode, see Section 6) which furthermore decreases the computational complexity to $\mathcal{O}(\log k \log M))$. Please note, that bidders still need to submit k bid values.

However, MPC based on homomorphic encryption is currently only possible for factorization based encryption schemes like Paillier encryption [38]. In contrast to discrete logarithm based schemes, the joint generation of secret keys needed for such schemes is very inefficient [39,40,41], especially when requiring full privacy. There is (yet) no general MPC scheme based on ElGamal encryption.

6 Conclusion

We presented a novel cryptographic auction protocol where bidders jointly compute the auction outcome in a constant number of rounds (three when assuming the random oracle model). The price we pay for round complexity that does neither depend on the number of bidders n nor on the number of possible bids k is computational complexity that is linear in k. However, experimental results indicate that the computational amount and message sizes are manageable in many realistic settings, despite its linearity in k.

The protocol complies with the highest standard of privacy possible: it is safe for a single bidder no matter how many of the other participants collude. The only agent being able to discover who won the auction besides the concerned bidders is the seller. We are not aware of any auction protocol, that achieves a similar level of privacy. Only computationally unbounded adversaries can uncover information. When using verifiable secret sharing instead of homomorphic encryption (like in [32]), only bid statistics are revealed to less than $n-1$ unbounded adversaries. As the protocol is publicly verifiable, malicious bidders that do not follow the protocol will be detected immediately and can be excluded from the set of bidders.

Table 1. Protocol complexity (computation per bidder)

Price	Rounds	Computation (exponentiations)
Private	$\mathcal{O}(1)$	INT: $\mathcal{O}(n^2k)$, PRE: $\mathcal{O}(nkT)$, DET: $\mathcal{O}(n^2kM)$
Public	$\mathcal{O}(1)$	INT: $\mathcal{O}(nk)$, PRE: $\mathcal{O}(kT)$, DET: $\mathcal{O}(nkM)$

n: bidders, k: prices/possible bids, M: units to be sold, T: ties

Table 1 shows the complexity of the protocol. When the selling price does not need to be protected ("public price"), the computational complexity can be reduced by just computing *one* value for all bidders that indicates the selling price p_w. Winning bidders can prove their claims to the seller by showing that $(\alpha_{iw}, \beta_{iw})$ is an encryption of Y. However, winning bidders are able to remain silent if they dislike the selling price (violating non-repudiation) in public-price mode. This could be circumvented by forcing all bidders to open their commitments for the selling price, thus proving to the seller whether they won or lost.

The protocol can be easily adapted to execute first-price or ascending (e.g. English) auctions. The latter might be useful in common-value scenarios where valuations interdepend. In the future, we intend to apply the presented techniques to solve tractable instances of combinatorial auctions like general multi-unit or linear-good auctions while maintaining full privacy.

References

1. Vickrey, W.: Counter speculation, auctions, and competitive sealed tenders. Journal of Finance **16** (1961) 8–37
2. Rothkopf, M., Teisberg, T., Kahn, E.: Why are Vickrey auctions rare? Journal of Political Economy **98** (1990) 94–109
3. Rothkopf, M., Harstad, R.: Two models of bid-taker cheating in Vickrey auctions. Journal of Business **68** (1995) 257–267
4. Sandholm, T.: Limitations of the Vickrey auction in computational multiagent systems. In: Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS), Menlo Park, CA, AAAI Press (1996) 299–306
5. Brandt, F., Weiß, G.: Vicious strategies for Vickrey auctions. In Müller, J., Andre, E., Sen, S., Frasson, C., eds.: Proceedings of the 5th International Conference on Autonomous Agents, ACM Press (2001) 71–72
6. Brandt, F., Weiß, G.: Antisocial agents and Vickrey auctions. In Meyer, J.J.C., Tambe, M., eds.: Intelligent Agents VIII. Volume 2333 of Lecture Notes in Artificial Intelligence., Springer (2001) 335–347 Revised papers from the 8th Workshop on Agent Theories, Architectures and Languages.
7. Brandt, F.: Cryptographic protocols for secure second-price auctions. In Klusch, M., Zambonelli, F., eds.: Cooperative Information Agents V. Volume 2182 of Lecture Notes in Artificial Intelligence., Springer (2001) 154–165
8. Wurman, P., Walsh, W., Wellman, M.: Flexible double auctions for electronic commerce: Theory and implementation. Decision Support Systems **24** (1998) 17–27
9. Franklin, M., Reiter, M.: The design and implementation of a secure auction service. IEEE Transactions on Software Engineering **22** (1996) 302–312
10. Abe, M., Suzuki, K.: M+1-st price auction using homomorphic encryption. In: Proceedings of the 5th International Conference on Public Key Cryptography (PKC). Volume 2274 of Lecture Notes in Computer Science., Springer (2002) 115–224
11. Abe, M., Suzuki, K.: Receipt-free sealed-bid auction. In: Proceedings of the 1st Information Security Conference (ISC). Volume 2433 of Lecture Notes in Computer Science. (2002) 191–199
12. Baudron, O., Stern, J.: Non-interactive private auctions. In: Proceedings of the 5th Annual Conference on Financial Cryptography (FC). (2001) 300–313
13. Cachin, C.: Efficient private bidding and auctions with an oblivious third party. In: Proceedings of the 6th ACM Conference on Computer and Communications Security. (1999) 120–127
14. Harkavy, M., Tygar, J., Kikuchi, H.: Electronic auctions with private bids. In: Proceedings of the 3rd USENIX Workshop on Electronic Commerce. (1998) 61–74
15. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In: Proceedings of the 6th Asiacrypt Conference. (2000) 162–177
16. Kikuchi, H.: (M+1)st-price auction protocol. In: Proceedings of the 5th Annual Conference on Financial Cryptography (FC). Volume 2339 of Lecture Notes in Computer Science., Springer (2001) 351–363
17. Kikuchi, H., Harkavy, M., Tygar, J.: Multi-round anonymous auction protocols. In: Proceedings of the 1st IEEE Workshop on Dependable and Real-Time E-Commerce Systems. (1998) 62–69
18. Kikuchi, H., Hotta, S., Abe, K., Nakanishi, S.: Resolving winner and winning bid without revealing privacy of bids. In: Proceedings of the International Workshop on Next Generation Internet (NGITA). (2000) 307–312

19. Kudo, M.: Secure electronic sealed-bid auction protocol with public key cryptography. IEICE Trans. Fundamentals **E81-A** (1998)
20. Lipmaa, H., Asokan, N., Niemi, V.: Secure Vickrey auctions without threshold trust. In Blaze, M., ed.: Proceedings of the 6th Annual Conference on Financial Cryptography (FC). Volume 2357 of Lecture Notes in Computer Science., Springer (2002)
21. Naor, M., Pinkas, B., Sumner, R.: Privacy preserving auctions and mechanism design. In: Proceedings of the 1st ACM Conference on Electronic Commerce. (1999) 129–139
22. Sako, K.: An auction protocol which hides bids of losers. In: Proceedings of the 3rd International Conference on Public Key Cryptography (PKC). Volume 1751 of Lecture Notes in Computer Science., Springer (2000) 422–432
23. Sakurai, K., Miyazaki, S.: A bulletin-board based digital auction scheme with bidding down strategy – towards anonymous electronic bidding without anonymous channels nor trusted centers. In: Proceedings of the International Workshop on Cryptographic Techniques and E-Commerce. (1999) 180–187
24. Song, D., Millen, J.: Secure auctions in a publish/subscribe system. Available at http://www.csl.sri.com/users/millen/ (2000)
25. Viswanathan, K., Boyd, C., Dawson, E.: A three phased schema for sealed bid auction system design. In: Proceedings of the Australasian Conference for Information Security and Privacy (ACISP). Lecture Notes in Computer Science (2000) 412–426
26. Watanabe, Y., Imai, H.: Reducing the round complexity of a sealed-bid auction protocol with an off-line TTP. In: Proceedings of the 7th ACM Conference on Computer and Communications Security, ACM Press (2000) 80–86
27. Sakurai, K., Miyazaki, S.: An anonymous electronic bidding protocol based on a new convertible group signature scheme. In: Proceedings of the 5th Australasian Conference on Information Security and Privacy (ACISP2000). Lecture Notes in Computer Science (2000)
28. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC). (1988) 1–10
29. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology **4** (1991) 161–174
30. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Advances in Cryptology – Proceedings of the 12th Annual International Cryptology Conference (CRYPTO). Volume 740 of Lecture Notes in Computer Science., Springer (1992) 3.1–3.6
31. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in Cryptology – Proceedings of the 14th Annual International Cryptology Conference (CRYPTO). Volume 893 of Lecture Notes in Computer Science., Springer (1994) 174–187
32. Brandt, F.: A verifiable, bidder-resolved auction protocol. In Falcone, R., Barber, S., Korba, L., Singh, M., eds.: Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies (Special Track on Privacy and Protection with Multi-Agent Systems). (2002) 18–25
33. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31** (1985) 469–472

34. Tsiounis, Y., Yung, M.: On the security of ElGamal-based encryption. In: Proceedings of the 1st International Workshop on Practice and Theory in Public Key Cryptography (PKC). Volume 1431 of Lecture Notes in Computer Science., Springer (1998) 117–134
35. Brandt, F.: Social choice and preference protection – Towards fully private mechanism design. In: Proceedings of the 4th ACM Conference on Electronic Commerce, ACM Press (2003) to appear.
36. Brandt, F.: Private public choice. Technical Report FKI-247-03, Department for Computer Science, Technical University of Munich (2003)
37. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Advances in Cryptology – Proceedings of the 18th Eurocrypt Conference. Volume 2045 of Lecture Notes in Computer Science., Springer (2001)
38. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology – Proceedings of the 16th Eurocrypt Conference. Volume 1592 of Lecture Notes in Computer Science., Springer (1999) 223–238
39. Algesheimer, J., Camenisch, J., Shoup, V.: Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In: Advances in Cryptology – Proceedings of the 22th Annual International Cryptology Conference (CRYPTO). Volume 2442 of Lecture Notes in Computer Science., Springer (2002) 417–432
40. Boneh, D., Franklin, M.: Efficient generation of shared RSA keys. In: Advances in Cryptology – Proceedings of the 17th Annual International Cryptology Conference (CRYPTO). Volume 1294., Springer (1997) 425–439
41. Damgård, I., Koprowski, M.: Practical threshold RSA signatures without a trusted dealer. In: Advances in Cryptology – Proceedings of the 18th Eurocrypt Conference. Volume 2045 of Lecture Notes in Computer Science., Springer (2001) 152–165

Secure Generalized Vickrey Auction Using Homomorphic Encryption

Koutarou Suzuki[1] and Makoto Yokoo[2]

[1] NTT Information Sharing Platform Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan
koutarou@isl.ntt.co.jp
[2] NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
www.kecl.ntt.co.jp/csl/ccrg/members/yokoo/
yokoo@cslab.kecl.ntt.co.jp

Abstract. Combinatorial auctions have recently attracted the interest of many researchers due to their promising applications such as the spectrum auctions recently held by the FCC. In a combinatorial auction, multiple items with interdependent values are sold simultaneously and bidders are allowed to bid on any combination of items. The Generalized Vickrey Auction (GVA) can handle combinatorial auctions and has several good theoretical characteristics. However, GVA has not yet widely used in practice due to its vulnerability to fraud by the auctioneers.
In this paper, to prevent such fraud, we propose a secure Generalized Vickrey Auction scheme where the result of the auction can be computed while the actual evaluation values of each bidder are kept secret.

Keywords: Generalized Vickrey Auction, combinatorial auction, homomorphic encryption, mechanism design, game-theory.

1 Introduction

Combinatorial auctions have recently attracted considerable attention [8,14,18, 19,29,30,38,39]. An extensive survey is presented in [7]. In contrast with conventional auctions that sell a single item at a time, combinatorial auctions sell multiple items with interdependent values simultaneously and allow the bidders to bid on any combination of items.

In a combinatorial auction, a bidder can express complementary/substitutable preferences over multiple bids. For example, in the Federal Communications Commission (FCC) spectrum auction [21], a bidder could indicate his desire for licenses covering adjoining regions simultaneously (i.e., these licenses are complementary), while being indifferent as to which particular channel was awarded (channels are substitutable). By supporting the complementary/substitutable preferences, we can increase the participants' utility and the revenue of the seller.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 239–249, 2003.
© Springer-Verlag Berlin Heidelberg 2003

The Generalized Vickrey Auction (GVA) [34], which is also known as the Vickrey-Clarke-Groves (VCG) mechanism, is a generalized version of the well-known Vickrey auction [35] and one instance of Clarke-Groves mechanism [6,9]. GVA can handle combinatorial auctions and has the following good theoretical characteristics.

Incentive Compatibility: For each bidder, truthfully declaring his/her evaluation values is the dominant strategy, i.e., the optimal strategy regardless of the actions of other bidders.

Pareto Efficiency: If all bidders take the dominant strategy (i.e., at the dominant strategy equilibrium), the social surplus, i.e., the sum of all bidders' utilities, is maximized.

Individual Rationality: No bidder suffers any loss by participating the auction.

Also, under certain assumptions, we can show that only the GVA can satisfy all of these properties while maximizing the expected revenue of the auctioneer [16].

Although GVA has these good theoretical characteristics, even its simplest form, i.e., the Vickrey auction, is not yet widely used. As discussed in [26], the main difficulty of using the Vickrey auction is its vulnerability to an insincere auctioneer. For example, if the highest bid is \$1,000 and the second highest bid is \$500, then, the payment of the winner becomes \$500. However, by fabricating a dummy bid at \$999, the auctioneer can increase his/her revenue to \$999.

Another difficulty is that the true evaluation value is sensitive information and a bidder may not want to reveal it [26]. For example, if a company wins in a public tender, then its bidding value, i.e., its true cost becomes public and the company may have difficulties in negotiating with sub-contractors.

This paper aims to provide a solution to these difficulties by developing a secure GVA scheme that utilizes homomorphic encryption. In the proposed scheme, the evaluation value of a bidder is represented by a vector of ciphertexts of homomorphic encryption, which enables the auctioneer to find the maximum value and add a constant securely, while the actual evaluation values are kept secret.

In contrast to the many works on sealed-bid auctions, see Section 1.1, there has not been paper on secure GVA with the following remarkable exception [22].

The rest of this paper is organized as follows. In Section 1.1, we discuss related works. In Section 2, we briefly explain GVA. In Section 3, we first explain the requirements of the auction, then introduce our secure GVA scheme and discuss its security and efficiency. In Section 4, we conclude the paper.

1.1 Related Works

Many papers consider secure sealed-bid auction. Kikuchi, Harkavy and Tygar presented an anonymous sealed-bid auction that uses encrypted vectors to represent bidding prices [13]. Harkavy, Tygar and Kikuchi proposed a Vickrey auction, where the bidding price is represented by polynomials that are shared by

auctioneers [10]. Kudo used a time server to realize sealed-bid auctions [17]. Cachin proposed a sealed-bid auction using homomorphic encryption and an oblivious third party [4]. Sakurai and Miyazaki proposed a sealed-bid auction in which a bid is represented by the bidder's undeniable signature of his bidding price [28]. Stubblebine and Syverson proposed an open-bid auction scheme that uses a hash chain technique [31]. Naor, Pinkas and Sumner realized a sealed-bid auction by combining Yao's secure computation with oblivious transfer [22]. Juels and Szydlo improved this scheme [11]. Sako proposed a sealed-bid auction in which a bid is represented by an encrypted message with a public key that corresponds to his bidding price [27]. Kobayashi, Morita and Suzuki proposed a sealed-bid auction that uses only hash chains [32,15]. Omote and Miyaji proposed a sealed-bid auction that is efficient [23]. Watanabe and Imai proposed a sealed-bid auction that utilizes chain of encryption [36]. Kikuchi proposed an $M+1$-st price auction, where the bidding price is represented by the degree of a polynomial shared by auctioneers [12]. Baudron and Stern proposed a sealed-bid auction based on circuit evaluation using homomorphic encryption [2]. Chida, Kobayashi and Morita proposed a sealed-bid auction with low round complexity [5]. Abe and Suzuki proposed a $M+1$-st price auction using homomorphic encryption [1]. Lipmaa, Asokan and Niemi proposed a $M+1$-st price auction without threshold trust [20]. Omote and Miyaji proposed a $M+1$-st price auction using p-th residue problem [24]. Suzuki and Yokoo proposed a combinatorial auction that uses secure dynamic programming [33,40]. Brandt proposed a $M+1$-st price auction where bidders compute the result by themselves [3].

In all of these schemes, however, the GVA has not been treated with a remarkable exception [22]. Naor, Pinkas and Sumner [22] proposed a general method for executing any auction including combinatorial auction based on a technique called the garbled circuit [37]. This method does not require interactive communications among multiple evaluators. However, to design a combinatorial circuit to implement GVA is still an open problem and the obtained circuit can be prohibitively large. Therefore, to develop a special purpose scheme for the GVA would be worthwhile.

2 Generalized Vickrey Auction

In this section, we briefly explain the Generalized Vickrey Auction (GVA) that is a type of sealed-bid combinatorial auction. Details of the protocol are as follows.

Let $B = \{1, 2, ..., i, ..., b\}$ be the set of b bidders, $G = \{1, 2, ..., j, ..., g\}$ be the set of g goods, $P = \{1, 2, ..., k, ..., p\}$ be the set of possible evaluation values. Let $B^G = \{A : G \to B\}$ be the set of b^g allocations of goods G to bidders B. (Notice that the allocation where some goods are not assigned to any bidder can be handled by introducing dummy bidder.)

Bidding: Each bidder i bids his/her evaluation value (function) $b_i : B^G \to P$, i.e., evaluation values for all allocations, in sealed manner.

Opening: The auctioneer reveals each sealed evaluation value and computes the allocation $A^* \in B^G$ that attains maximum $max_{A \in B^G} \sum_i b_i(A)$ of sum of all

evaluation values, i.e., social surplus. This allocation is the result of the auction, so the goods are sold according to this allocation. The auctioneer then computes the payment p_x of bidder x by the following formula

$$p_x = \sum_{i \neq x} b_i(A^*_{\sim x}) - \sum_{i \neq x} b_i(A^*),$$

where $A^*_{\sim x} \in B^G$ is the allocation that attains maximum $max_{A \in B^G} \sum_{i \neq x} b_i(A)$. Bidder x makes payment p_x for goods sold to him/her according to allocation A^*.

We show that GVA achieves dominant-strategy incentive compatibility, i.e., for each bidder, truthfully declaring his/her evaluation values is the dominant strategy, i.e., an optimal strategy regardless of the actions of other bidders. Assuming that each bidder has quasi-linear utility, i.e., "utility" = "true evaluation value" − "payment", bidder x has the following utility $u_x(b_x)$ as a function of his/her evaluation value b_x

$$u_x(b_x) = v_x(A^*) - p_x = v_x(A^*) + \sum_{i \neq x} b_i(A^*) - \sum_{i \neq x} b_i(A^*_{\sim x}),$$

where $v_x : B^G \to P$ is the true evaluation value (function) of bidder x for goods allocations. The third term does not depend on evaluation value b_x, so bidder x wants to maximize the sum of the first term and the second term. Since A^* is determined so as to maximize $\sum_i b_i$, the sum $v_x + \sum_{i \neq x} b_i$ can be maximized by setting $b_x := v_x$. Thus the strategy of bidding the true evaluation value v_x is the dominant strategy regardless of the actions of other bidders. Notice that the case of $g = 1$ is the conventional Vickrey auction (second-price auction). It is clear that if all bidders truthfully declare their evaluation values, the social surplus, i.e., the sum of all bidders' utilities is maximized by allocation A^*, i.e., GVA satisfies Pareto efficiency.

Example: To make the auction comprehensible, we provide the following small example where $B = \{1, 2\}$ and $G = \{1, 2\}$. This means

$$B^G = \{A_1 = (\{1, 2\}, \{\}), A_2 = (\{1\}, \{2\}), A_3 = (\{2\}, \{1\}), A_4 = (\{\}, \{1, 2\})\},$$

where, e.g., $(\{1, 2\}, \{\})$ means goods 1 and 2 are allocated to bidder 1 and no goods to bidder 2. Evaluation values b_1 and b_2 of bidders 1 and 2 are

$$b_1 = (3, 2, 2, 0) \quad \text{and} \quad b_2 = (0, 2, 0, 3)$$

where $f = (a_1, a_2, a_3, a_4)$ means $f(A_1) = a_1, f(A_2) = a_2, f(A_3) = a_3, f(A_4) = a_4$. We then have

$$b_1 + b_2 = (3, 4, 2, 3).$$

We then have $max_{A \in B^G} \sum_{i=1,2} b_i(A) = 4$ and $A^* = A_2$ and

$$p_1 = b_2(A_4) - b_2(A_2) = 3 - 2 = 1 \quad \text{and} \quad p_2 = b_1(A_1) - b_1(A_2) = 3 - 2 = 1.$$

So bidder 1 buys goods 1 at price $p_1 = 1$ and bidder 2 buys goods 2 at price $p_2 = 1$.

3 Secure Generalized Vickrey Auction

In this section, we first explain the requirements and preliminaries. We then introduce our secure GVA scheme and discuss its security and efficiency.

3.1 Requirements

As discussed in Section 1, to prevent the fraud of the auctioneer and the leakage of sensitive information, declared evaluation values should be kept secret.

Secrecy of evaluation value: Only the result of the auction, i.e., allocation of goods A^* and payments p_i should be made public while all evaluation values b_i must be kept secret even from the auctioneer.

3.2 Preliminaries

Let E be a probabilistic public key encryption that provides indistinguishability, homomorphic property, and randomizability. The homomorphic property means that $E(a)E(b) = E(ab)$, and the randomizability means that one can compute a randomized ciphertext $E'(m)$ only from the original ciphertext $E(m)$, i.e. without knowing either the decryption key or the plaintext. For instance, ElGamal encryption or Paillier encryption [25] have the properties desired, so our auction scheme can be built on these encryption schemes.

To compute the result of GVA securely, we have to find the maximum of prices and add prices without revealing the prices themselves. First, we explain the representation of the price that makes these tasks feasible.

Vector representation: We represent price w $(1 \leq w \leq n)$ by vector $\boldsymbol{e}(w)$ of ciphertexts

$$\boldsymbol{e}(w) = (e_1, \ldots, e_n) = (\underbrace{E(z), ..., E(z)}_{w}, \underbrace{E(1), ..., E(1)}_{n-w}),$$

where $E(1)$ and $E(z)$ denote the encryption of 1 and common public element $z(\neq 1)$, respectively. Here, $\mathrm{ord}(z)$ and n are chosen large enough. Because of the indistinguishability of E, we cannot determine w without decrypting each element.

Find the maximum: We can find the maximum of encrypted prices $\boldsymbol{e}(w_i) = (e_{1,i}, \ldots, e_{n,i})$ without leaking any information about the prices that are not the maximum as follows. Consider the componentwise product of all vectors

$$\prod_i \boldsymbol{e}(w_i) = (\prod_i e_{1,i}, \ldots, \prod_i e_{n,i}).$$

Observe that, due to the homomorphic property, the j-th component of this vector has the following form

$$c_j = \prod_i e_{j,i} = E(z^{S(j)})$$

where $S(j) = \#\{i \mid j \leq w_i\}$ is the number of prices that are equal or greater than j. Notice that $S(j)$ monotonically reduces as j increases. To find the maximum of these prices, we decrypt c_j and check whether decryption $D(c_j)$ is equal to 1 or not from $j = n$ to $j = 1$ until we find the largest j s.t. $D(c_j) \neq 1$. This j is equal to $\max_i\{w_i\}$, i.e., the maximum of the prices.

Add a constant: We can add a constant c to encrypted price $\boldsymbol{e}(w) = (e_1, \ldots, e_n)$ without learning w. By shifting and randomizing $\boldsymbol{e}(w)$, we can obtain

$$\boldsymbol{e'}(w+c) = (\underbrace{E(z), \ldots, E(z)}_{c}, e'_1, \ldots, e'_{n-c})$$

where e'_j is a randomization of ciphertext e_j. Due to randomization, one can obtain no information about constant c from $\boldsymbol{e}(w)$ and $\boldsymbol{e'}(w+c)$. Note that we can perform these operations without decrypting $\boldsymbol{e}(w)$ nor learning w.

By representing prices using the vector representation, we can find the maximum of prices and add prices without learning the prices themselves, thus we can securely compute the result of GVA.

3.3 Proposed Scheme

The proposed scheme is as follows.

Preparation: Let $B = \{1, 2, ..., i, ..., b\}$ be the set of b bidders, $G = \{1, 2, ..., j, ..., g\}$ be the set of g goods, $P = \{1, 2, ..., k, ..., p\}$ be the set of possible evaluation values. Let $B^G = \{A : G \to B\}$ be the set of b^g allocations of goods G to bidders B. For simplicity of description, we denote by $\boldsymbol{E}(f) = (\ \boldsymbol{e}(f(A))\ )_{A \in B^G}$ the vector representation of evaluation value function $f : B^G \to P$. There is an auctioneer that computes the results of the auction. The auctioneer is implemented by plural servers to prevent a malicious auctioneer from learning the evaluation value. Indeed, the decryption to find the maximum of prices and the addition of random mask constant r in the following protocol are performed in a distributed manner by these servers. For simplicity, we do not describe this explicitly.

The auctioneer generates his/her secret and public key of homomorphic encryption E, and publishes the set of possible evaluation values P of the auction, the public key of homomorphic encryption E, and element $z(\neq 1)$.

Bidding: Each bidder x decides his/her evaluation value function $b_x : B^G \to P$, i.e., evaluation values for all allocations.

The auctioneer makes $b+1$ representations $\boldsymbol{E}_0 = \boldsymbol{E}(O), \boldsymbol{E}_1 = \boldsymbol{E}(O), ..., \boldsymbol{E}_b = \boldsymbol{E}(O)$ of constant zero function $O(A) = 0$. Each bidder x adds his/her evaluation value function b_x to representations $\boldsymbol{E}_0, \boldsymbol{E}_1, ..., \boldsymbol{E}_{x-1}, \boldsymbol{E}_{x+1}, ..., \boldsymbol{E}_b$ except x-th representation $\boldsymbol{E}_x$, while keeping evaluation value function b_x secret. After all bidders have done this, the auctioneer has

$$\boldsymbol{E}_0 = \boldsymbol{E}(\sum_i b_i), \quad \boldsymbol{E}_x = \boldsymbol{E}(\sum_{i \neq x} b_i) \quad x = 1, 2, ..., b.$$

Opening: First, the auctioneer computes $\boldsymbol{E}(\sum_i b_i + R)$ from $\boldsymbol{E}_0$ by adding random constant function $R(A) = r$ to mask the value. By using $\boldsymbol{E}(\sum_i b_i + R)$, the auctioneer finds masked maximum

$$m = max_{A \in B^G}((\sum_i b_i + R)(A)) = max_{A \in B^G}((\sum_i b_i)(A)) + r.$$

The auctioneer then decrypts the m-th element of vector $\boldsymbol{e}((\sum_i b_i + R)(A))$ for all allocations $A \in B^G$, and checks whether the decryption is equal to z or not. If it is equal to z at allocation $A^* \in B^G$, the auctioneer finds that allocation A^* is the one that maximizes sum $\sum_i b_i$ of all evaluation values. The allocation is the result of the auction, so the goods are sold according to allocation A^*.

The auctioneer then computes the payment p_x of bidder x as follows. First, the auctioneer computes $\boldsymbol{e}((\sum_{i \neq x} b_i)(A^*) + r')$ from component $\boldsymbol{e}((\sum_{i \neq x} b_i)(A^*))$ of $\boldsymbol{E}_x$ by adding random constant r' to mask the value. By using this, the auctioneer finds masked maximum $\sum_{i \neq x} b_i(A^*) + r'$.

Next, the auctioneer computes $\boldsymbol{E}(\sum_{i \neq x} b_i + R')$ from $\boldsymbol{E}_x$ by adding random constant function $R'(A) = r'$ to mask the value. By using this, the auctioneer finds masked maximum $max_{A \in B^G}((\sum_{i \neq x} b_i)(A)) + r'$. (This is equal to $\sum_{i \neq x} b_i(A^*_{\sim x}) + r'$ by definition of $A^*_{\sim x}$.)

The auctioneer then finds the payment $p_x = \sum_{i \neq x} b_i(A^*_{\sim x}) - \sum_{i \neq x} b_i(A^*)$ by subtracting these masked values. Bidder x pays his/her payment p_x for goods sold to him/her according to allocation A^*.

Example : To make the protocol comprehensible, we explain the case of the example in section 2 where $B = \{1, 2\}$ and $G = \{1, 2\}$.

The auctioneer makes $\boldsymbol{E}_0, \boldsymbol{E}_1, \boldsymbol{E}_2 = (\boldsymbol{e}(0), \boldsymbol{e}(0), \boldsymbol{e}(0), \boldsymbol{e}(0))$, then bidder 1 adds evaluation values $b_1 = (3, 2, 2, 0)$ to $\boldsymbol{E}_0, \boldsymbol{E}_2$ and bidder 2 adds evaluation values $b_2 = (0, 2, 0, 3)$ to $\boldsymbol{E}_0, \boldsymbol{E}_1$. We have

$$\boldsymbol{E}_0 = (\boldsymbol{e}(3), \boldsymbol{e}(4), \boldsymbol{e}(2), \boldsymbol{e}(3)), \boldsymbol{E}_1 = (\boldsymbol{e}(0), \boldsymbol{e}(2), \boldsymbol{e}(0), \boldsymbol{e}(3)), \boldsymbol{E}_2 = (\boldsymbol{e}(3), \boldsymbol{e}(2), \boldsymbol{e}(2), \boldsymbol{e}(0)).$$

The auctioneer adds random constant function $R(A) = r = 2$ to $\boldsymbol{E}_0$ to yield

$$\boldsymbol{E}(\sum_i b_i + R) = (\boldsymbol{e}(3+2), \boldsymbol{e}(4+2), \boldsymbol{e}(2+2), \boldsymbol{e}(3+2)),$$

takes the componentwise product $\boldsymbol{e}(3+2) \cdot \boldsymbol{e}(4+2) \cdot \boldsymbol{e}(2+2) \cdot \boldsymbol{e}(3+2)$, and then decrypts this to find $max_{A \in B^G}((\sum_{i=1,2} b_i)(A)) + r = 4+2$. The auctioneer then decrypts the $(4+2)$-th element of $\boldsymbol{e}(3+2), \boldsymbol{e}(4+2), \boldsymbol{e}(2+2), \boldsymbol{e}(3+2)$ to find $A^* = A_2$.

The auctioneer adds random constant $r' = 1$ to the 2-nd component $\boldsymbol{e}(2)$ of $\boldsymbol{E}_1$ to yield

$$\boldsymbol{e}((\sum_{i \neq x} b_i)(A^*) + r') = \boldsymbol{e}(2+1)$$

and decrypts $\boldsymbol{e}(2+1)$ to find $(\sum_{i \neq 1} b_i)(A^*) + r' = b_2(A_2) + r' = 2 + 1$.

The auctioneer adds random constant function $R'(A) = r' = 1$ to $\boldsymbol{E}_1$ to yield

$$\boldsymbol{E}(\sum_{i \neq 1} b_i + R') = (\boldsymbol{e}(0+1), \boldsymbol{e}(2+1), \boldsymbol{e}(0+1), \boldsymbol{e}(3+1)),$$

takes the componentwise product $\boldsymbol{e}(0+1) \cdot \boldsymbol{e}(2+1) \cdot \boldsymbol{e}(0+1) \cdot \boldsymbol{e}(3+1)$, and then decrypts this to find $(\sum_{i \neq 1} b_i)(A^*_{\sim 1}) + r' = b_2(A_4) + r' = 3 + 1$.

We then have $p_1 = (3-1)-(2-1) = 3-2 = 1$. We also compute $p_2 = 3-2 = 1$ in the same manner.

3.4 Security

We discuss here the security issue of our auction. First, because of the indistinguishability of encryption E, we can learn nothing about price p from its representation $\boldsymbol{e}(p)$ without decrypting each element.

In our scheme, the auctioneer is implemented by plural servers to prevent a malicious auctioneer from learning the evaluation values. Since the decryption to find the maximum of evaluation values is performed in a distributed manner by these servers, no malicious auctioneer can decrypt illegally to learn about evaluation values. Since the addition of random mask constant r is also performed in a distributed manner by these servers, no malicious auctioneer can learn random mask constant r. Therefore, our scheme can hide most of the information of the evaluation values.

In our scheme, however, some information is leaked, since a random mask value is added over the integers, unlike the perfect secrecy achieved by addition in a finite group. So our scheme leaks some information besides the result of the auction, i.e., allocation of goods A^* and payments $p_1, p_2, ..., p_b$.

3.5 Efficiency

We discuss here the communication and computational complexity of our auction. Here, the number of auctioneers, bidders, goods, and possible evaluation values, are denoted by a, b, g, and p, respectively.

Table 1 shows the communication pattern, the number of communication rounds, and the volume per communication round in our scheme. The communication cost is linear against the number of evaluation values p, so it may impose a heavy cost for a large range of evaluation values, the same as most of the existing schemes. The cost is also exponential against g, so it may impose a heavy cost for a large number of goods, however this is inevitable for combinatorial auction. Since communications from bidder B_j to auctioneer A_i is required only in the bidding phase, our scheme achieves the "bid and go" concept.

Table 2 shows the computational complexity per bidder and per all auctioneers in our scheme. The complexity of each bidder and all auctioneers is linear against the number of evaluation values p, so it may impose a heavy cost for a large range of evaluation values, the same as most of the existing schemes. The cost is also exponential against g, so it may impose a heavy cost for a large number of goods, however this is inevitable for combinatorial auction.

Table 1. The communication complexity of our scheme.

	pattern	round	volume
bidding	bidder $\leftrightarrow$ auctioneer	b	$O(b \times b^g \times bp)$
opening	auctioneer $\leftrightarrow$ auctioneer	$a \times bp$	$O(b \times b^g)$

Table 2. The computational complexity of our scheme.

	computational complexity
one bidder	$O(b \times b^g \times bp)$
auctioneers	$O(a \times b \times b^g \times bp)$

Our proposed scheme requires that each bidder declare his/her evaluation values for all b^g possible allocations. This is inevitable to implement GVA for general cases. However, for many auctions in the real world, we can assume the following two conditions.

- No allocative externality, i.e., each bidder only concerns with the goods that are allocated to him/her and he/she is indifferent to the allocations of other bidders.
- Free disposal, i.e., goods can be discarded without any cost.

In this case, a bidder needs to declare his/her evaluation values only for the set of goods in which he/her is interested, so we can reduce the number of allocations from b^g to 2^g. This means that auctions involving a large number of bidders are feasible.

4 Conclusion

We proposed a secure Generalized Vickrey Auction scheme that hides the evaluation values of bidders by utilizing homomorphic encryption. In our scheme, the evaluation value of a bidder is represented by a vector of ciphertexts of homomorphic encryption, which enables the auctioneer to find the maximum value and add a constant securely, while the actual evaluation values are kept secret.

Finally, we mention that we can make our scheme verifiable and secure against active adversaries by using zero-knowledge proof of equality of logarithms and OR-proof technique.

Acknowledgments. The authors thank the anonymous reviewers for their useful comments.

References

1. Masayuki Abe and Koutarou Suzuki. M+1-st price auction using homomorphic encryption. *Proceedings of Public Key Cryptography 2002*, pages 115–124, 2002. Springer LNCS 2274.

2. O. Baudron and J. Stern. Non-interactive private auctions. *Proceedings of Financial Cryptography 2001*, pages 364–377, 2001. Springer LNCS 2339.
3. Felix Brandt. Fully private auctions in a constant number of rounds. *Proceedings of Financial Cryptography 2003*, 2003.
4. C. Cachin. Efficient private bidding and auctions with an oblivious third party. *Proceedings of 6th ACM Conference on Computer and Communications Security*, pages 120–127, 1999.
5. K. Chida, K. Kobayashi, and H. Morita. Efficient sealed-bid auctions for massive numbers of bidders with lump comparison. *Proceedings of Information Security Conference 2001*, pages 408–419, 2001. Springer LNCS 2200.
6. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 2:19–33, 1971.
7. Sven de Vries and Rakesh V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, forthcoming.
8. Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computation complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.
9. Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
10. M. Harkavy, J. D. Tygar, and H. Kikuchi. Electronic auctions with private bids. *Proceedings of Third USENIX Workshop on Electronic Commerce*, pages 61–74, 1998.
11. A. Juels and M. Szydlo. A two-server, sealed-bid auction protocol. *Proceedings of Financial Cryptography 2002*, pages 72–86, 2002. Springer LNCS 2357.
12. H. Kikuchi. (m+1)st-price auction protocol. *Proceedings of Financial Cryptography 2001*, pages 351–363, 2001. Springer LNCS 2339.
13. H. Kikuchi, M. Harkavy, and J. D. Tygar. Multi-round anonymous auction protocols. *Proceedings of first IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, pages 62–69, 1998.
14. Paul Klemperer. Auction theory: A guide to the literature. *Journal of Economics Surveys*, 13(3):227–286, 1999.
15. Kunio Kobayashi, Hikaru Morita, Koutarou Suzuki, and Mitsuari Hakuta. Efficient sealed-bid auction by using one-way functions. *IEICE Trans. Fundamentals*, E84-A(1):289–294, 2001.
16. Vijay Krishna. *Auction Theory*. Academic Press, 2002.
17. M. Kudo. Secure electronic sealed-bid auction protocol with public key cryptography. *IEICE Trans. Fundamentals*, E81-A(1):20–27, 1998.
18. Daniel Lehmann, Liadan Ita O'Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auction. In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, pages 96–102, 1999.
19. Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 66–76, 2000.
20. H. Lipmaa, N. Asokan, and V. Niemi. Secure vickrey auctions without threshold trust. *Proceedings of Financial Cryptography 2002*, pages 87–101, 2002. Springer LNCS 2357.
21. John McMillan. Selling spectrum rights. *Journal of Economics Perspectives*, 8(3):145–162, 1994.
22. M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. *Proceedings of ACM conference on E-commerce 1999*, pages 129–139, 1999.

23. K. Omote and A. Miyaji. An anonymous auction protocol with a single non-trusted center using binary trees. *Proceedings of Information Security Workshop 2000*, pages 108–120, 2000. Springer LNCS 1975.
24. K. Omote and A. Miyaji. A second-price sealed-bid auction with the discriminant of the p-th root. *Proceedings of Financial Cryptography 2002*, pages 57–71, 2002. Springer LNCS 2357.
25. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Proceedings of EUROCRYPT '99*, pages 223–238, 1999.
26. M. H. Rothkopf, T. J. Teisberg, and E. P. Kahn. Why are vickrey auctions are rare. *Journal of Political Economy*, 98(1):94–109, 1990.
27. K. Sako. Universally verifiable auction protocol which hides losing bids. *Proceedings of Public Key Cryptography 2000*, pages 422–432, 2000. Springer LNCS 1751.
28. K. Sakurai and S. Miyazaki. A bulletin-board based digital auction scheme with bidding down strategy. *Proceedings of 1999 International Workshop on Cryptographic Techniques and E-Commerce*, pages 180–187, 1999.
29. Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
30. Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast combinatorial algorithm for optimal combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pages 1102–1108, 2001.
31. S. G. Stubblebine and P. F. Syverson. Fair on-line auctions without special trusted parties. *Proceedings of Financial Cryptography 1999*, pages 230–240, 1999. Springer LNCS 1648.
32. Koutarou Suzuki, Kunio Kobayashi, and Hikaru Morita. Efficient sealed-bid auction using hash chain. *Proceedings of International Conference Information Security and Cryptology 2000*, pages 183–191, 2000. Springer LNCS 2015.
33. Koutarou Suzuki and Makoto Yokoo. Secure combinatorial auctions by dynamic programming with polynomial secret sharing. *Proceedings of Financial Cryptography 2002*, pages 44–56, 2002. Springer LNCS 2357.
34. Hal R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First Usenix Workshop on Electronic Commerce*, 1995.
35. William Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
36. Yuji Watanabe and Hideki Imai. Reducing the round complexity of a sealed-bid auction protocol with an off-line ttp. *Proceedings of ACM Conference on Computer and Communications Security 2000*, pages 80–86, 2000.
37. A. C. Yao. How to generate and exchange secrets. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.
38. Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167–181, 2001.
39. Makoto Yokoo, Yuko Skurai, and Shigeo Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in internet auctions. *Games and Economic Behavior*, forthcoming.
40. Makoto Yokoo and Koutarou Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 112–119, 2002.

Trusted Computing Platforms: The Good, the Bad, and the Ugly

Moti Yung

Columbia University, New York.
moti@cs.columbia.edu

Abstract. This is a summary of a panel on Trusted Computing Platform Architectures that was held in Financial Cryptography 2003.

1 Introduction

Achieving trusted computing seems like the ultimate goal for security specialists. At the same time, this looks like a goal that can never be fully achieved due to the constantly increasing complexity of the computing environment. In this respect, the recent efforts of the TCPA (Trusted Computing Platform Alliance) and Palladium as major industry efforts (by major "industrial players") to provide a better trusted computing platform are both interesting and intriguing. Thus, it is natural that they have been, both, heavily supported by proponents in the community and heavily criticized by opponents of these ideas who have raised suspicions regarding the intent and the purpose of these platforms.

Given that TCPA and Palladium "trusted platform" activities have indeed raised many promises, questions and objections, and given that they further polarized the society of security experts, I thought it is interesting to confront the two camps in a panel discussion. The hope was that it could generate interesting interactions that may be important to future developments.

The subtitle of the panel represents some of the opinions about the technology. The "good" part is all good: cryptographic keys can be protected and used in a safe and secure way. The "bad" part is that corporate alliances and others can exert undue control and restrict users from employing the full power of general purpose computing, while the "ugly" part may refer to this technology being abused for marketing anti-competition purposes that may, in particular kill the open source efforts.

The panel's participants were: Dirk Kuhnman (HP), Paul Kocher (Cryptography Research) and Lucky Green [aka Marc Briceno] (independent security researcher). In this panel, we confronted proponent and opponents of trusted computing platform ideas, we motivated the conference participants to raise questions and debate and we hope we have raised more awareness regarding ways of use and abuse these platforms and the ideas behind them.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 250–254, 2003.
© Springer-Verlag Berlin Heidelberg 2003

2 Opinions

A Proponent View:

Dirk Kuhnman's position, which will be presented in the next paper, was that in certain communication scenarios "trusting yourself" is not good enough and these interactions need external source of trust. Abuses and objections, according to Dirk, are possible but are not part of the TCPA technology, rather they can come only from what can be built on top of it. His paper describes his position in full but let me say that in his presentation he supported the idea of TCPA and his conclusion was that instead of fighting this technology the community should focus on supporting the software and building something on it because openness is a necessary but not sufficient condition for creating a trustworthy TPCA.

A DRM Point of View:

Paul Kocher asked what is trustworthy computing: can you build a computer a user can trust? can you build a computer a networked anonymous person can trust? He stated the following: We are doing a terrible job of building machines worthy of a user trust because the complexity of a system is continuously increasing. It is no longer possible for a single person to know all things and all bits inside a machine. So even experts can no longer be certain.

Paul analyzed the media companies position and stated that they want to control high value commodity content on the machines of remote users. He asked: What are intellectual property rights and are they a good thing? Among technical people the notion of intellectual property rights is one that people meet with hostility. Intellectual property is the ability to dictate your own work. Intellectual property is property and by definition property is the right to exclude others from access. Refusing access to words means limiting speech rights, so intellectual property is a passionate debate because it is a conflict between the two core American rights: the right to property and the right to speech. Intellectual property owners have a right to remove the autonomy of users so they can be certain about the use of their content.

According to Paul, TCPA comes as a result of the failure to developed workable cryptographic solution to business requirements for intellectual property systems. In fact, practical applied research should solve Hollywood's problems or they will push for additional controls, and being in the position of control of the content they have power to push for it. The need for cryptographic solutions that adapt to changes in business models of content providers was the issue that Paul raised. He further believes that nothing technological is "totally wrong" and that the market will eventually decide the fate of this technology.

An Opponent View:

Lucky Green stated that he wants trusted computing very very badly since he knows he cannot trust his computer, but the trusted computing platforms are not what he wants. According to him when looking at the public statements about what the technology is intended to do, it is obvious that TCPA is supposed to make the PC the core of the home entertainment industry. He mentioned that the head of TCPA made five or six comments about how TCPA is absolutely not

for DRM, yet he also said: "There is certain content that owners will not make available on the PC platform; that is unacceptable and we will solve this problem one way or another." Thus, the conclusion is that the business objective of TPCA is DRM first and foremost and the goal is copy protection for the content and software companies.

The business analysis provided by Lucky is the following: TCPA is about defining the future of the PC. Anyone who would purchase a machine has done so by now. So how does one grow the market? According to the PC industry the market is saturated. Another market is the home entertainment center. At the center of the home entertainment system can be Sony 5.0 or something from Microsoft. Sony sells more consumer electronics than Microsoft has ever sold software. This market is giant and will be hotly contested. Microsoft believes that TCPA is the only way to win its coming battle with Sony for the heart of the home.

The conclusion that this analysis leads to is that the overall objective is to prevent user autonomy. This enforces three levels of access: (1) highest level access where you can see everything going on and you can know what is happening and you further know the state: this is reserved for owners of high value content not users! (2) user access, and (3) minimal access. Trustworthy computing now means that third parties can trust the computer to enforce rules in opposition to the desire of the users and companies can enforce which software can or cannot be used in handling the user's documents, say. The combination of the technology and DRM laws may mean legal monopoly of DRM based software (since reverse engineering DRM is illegal).

Lucky pointed out that the technology was advertised in various and changing ways: When soliciting members the proposal was "to enable secure boot." Within the working groups the purpose was "to enable DRM to serve the MPAA." Later the pitch was "to enable DRM for everybody." Now TPCA is "to eliminate all spam viruses and hacking." Next the architecture is likely to be pitched to the Office of Homeland Security.

He added possible countermeasures in order to reject TCPA: Demand owner override! The security of simple trusted system depends on the owner not having access to the cryptographic keys. If you do not have access to all the keys then you cannot control your own machines. He had no issue with the notion of intellectual property aspects because he does do not think they are relevant to the property debate. He does not care if content providers include various restrictions that content owners use to support intellectual property protection. What concerns him is that the content providers through the operating system providers are turning the general purpose machine into a machine with a platform for a back door that the user cannot control or close. The users do not have anymore root access to their own machine. This is alarming, because, as he put it, TCPA is designed to make computers less secure.

Dirk reacted to Lucky by saying that preventing root access on one's machine is a valid point. TCPA is about preventing root access while engaging in communications with another entity, while allowing access on your system at any other

time. This is about contractual agreements in communications situations. Now the honest guys do not want to do any harm but they cannot prove this fact; with TCPA they will be able to assure honest behavior. User override will be possible. Conceptually and technically TCPA clearly allows user override. If user override means key access - then lack of user access is very good because loss of user autonomy makes users trustworthy. Further, according to Dirk, migratable keys can come with different security classification. The issue of the complexity of override was debated as well.

3 Issues

The TCPA, according to its proponents, is a try to provide blocks that make it *possible* to design platforms that are as strong as necessary (or as weak as acceptable, including the ability to switch security off). The level of platform strength is ultimately that of the software system and infrastructure that make use of TCPA (or support it). TCPA defines nothing more but Protection Profiles for the chip and the way it has to be attached to the system hardware. This point of view is discussed in the next paper by Dirk who raises advantages and positive points regarding the TCPA technology.

Various issues beyond his personal opinion are discussed in Dirk's paper and the reader is encouraged to read it.

To present counterpoints, the following valid counter-arguments can be raised against Dirk's position (as discussed with Dirk):

1. There are fundamental problems with CA (Trusted Third Party) based approaches in case these CAs can be coerced (probably by law) into signing arbitrary key material, thus 'faking' trusted platform modules.
2. There are fundamental issues with black-box encryption, randomness and key generation (see Weis, R., Lucks, S. SANE 2002, Maastricht 2002), other problems involve Kleptographic attacks (Young A., Yung M., Crypto 96, Crypto 97).
3. What Dirk calls the "virtue of not knowing" (i.e., not 'holding' all private keys / not being able to see them all / not being able to generate all of them by the method of your choice) is considered by him to be a good thing. Others simply disagree, e.g., Whitfield Diffie thinks it isn't – apparently as a matter of principle.
4. It is at least debatable whether Trusted Computing TCPA/ Palladium fashion inherently favors mono-culture, as it is easier to manage a handful of trust metrics for a single, pre-compiled operating system than it is to do the same for a zillion different distributions, configurations and self-compiled operating systems versions.
5. Indeed, Lucky raised a valid point (according to Dirk) concerning intentions that might be considered as some of the the driving forces behind TCPA, Palladium etc. (e.g., DRM as also Paul assumes).

4 Conclusion

The debate, questions and answers and the discussion was a real learning experience. The debate was interesting as well. All three panelists like "trusted computing" but differ on the ways to achieve it; this reflects the state of the art of our understanding of secure platforms what they can achieve and in what ways we can assure their proper use.

I would like to thank the panelist who helped me as a moderator and the audience for their participation. I enjoyed being the moderator of this panel.

In my view the holders of the various opinion all play positive roles in this case. The *opponent side* points to threats and put the major companies involved in this effort (and the public) on alert, forcing the industry to be cautious regarding potential abuses, thus assuring that "the right path" is taken. The *proponent side*, on the other hand, tries to design the platform as strongly as possible to enable safe executions of applications that need the right level of protection.

On TCPA

Dirk Kuhlmann

HP Laboratories, Filton Road, Bristol BS34 8QZ, UK, dirk.kuhlmann@hp.com

Abstract. This document is an extended version of a presentation given at a Financial Cryptography 2003 panel on Trusted Computing Platform Architecture (TCPA).

1 Why Trusted Platforms May Be Hard to Avoid

1.1 Truceful Communication

In direct, unmediated interaction, it is comparatively easy to determine the communication context. When talking face-to-face, people normally figure out what is going on. For a start, they can almost always tell that they are talking to a person. They are usually aware whether they are expected to talk or e.g. to dance. Participants are either implicitly aware of both expectations and intentions of their respective counterparts, or they quickly infer them during a communication setup phase.

Direct symbolic interaction between humans is always accompanied by a set of contract-like agreements that regulate the communication protocol. Participants will implicitely agree on a common language, to apply the the rules of logic when making an argument, and not to threaten each other with physical violence. Entering a symbolic exchange presupposes the equivalent of a temporary 'truce' on the physical and protocol level.

Intermediation by traditional communication devices makes it slightly more difficult to determine contextual information. It does not change the situation fundamentally, though: written records on paper or voice communication over a telephone system typically still allow this to a wide extent, which is due to the very characteristics of these media. In particular, we do not have to consider the possibility that the medium itself actively participates in attempts to deceive us. Traditional communication media can therefore be regarded as mere tools that are transparent with respect to our goals and intentions.

This state of affairs, however, changes drastically once symbolic interaction is mediated by advanced information technology.

1.2 The Computer as Untrusted Third Party

It is only just becoming apparent to what extent using the computer as a communication device places trust in the proper function of the underlying technical systems today. Neither senders or receivers of bit streams have verifiable indicators about the integrity of their own systems, let alone those of their peers. In

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 255–269, 2003.
© Springer-Verlag Berlin Heidelberg 2003

essence, they have no other choice than to assume that the system will process, encode, transfer and decode their intentions correctly. Given the absence of indicators, the the proper function of the technical medium would be is taken on blind faith.

Advanced IT technology is full of components that have been produced by unknown third parties. We usually assume that these components are neutral with respect to our intentions although they can be 'hostile' in that they could interfere with our intentions. One might think of computer viruses and root kits here, but the problem is more fundamental, since every single bit of acquired software could be 'hostile' as well.

How can we underpin our assumption that all technical elements will respect (are ignorant of or transparent with regard to) – our own goals and intentions? We can't, as there is no trustworthy and verifiable data. We simply hope that the technology will operate according to the expectation of its owner. In reality, we are dealing with potential 'intelligent agents' with ideas and intentions beyond and quite possibly contrary to our own, without having indicators for estimating the risk whether this might actually be the case.

1.3 Expectations and Contracts

Today's computers include no mechanisms to provide trustworthy information about their own current system state. Even if they did, no communication mechanism that would allow to communicate this information trustworthy manner. In particular, this affects the communication setup phase for advanced IT systems and makes it radically different from traditional media.

As outlined above, the decision on whether or not to enter a communication with a peer in the first place relies on crucially on contextual information that allows estimates about expected behaviour. As current IT technology lacks mechanisms to monitor and report their state, there is no reliable context information that would allow to reflect the probability of it functioning properly. The foundation of the interaction would be informed decision on whether and to what extent a remote entity might honour the 'truce' (adhere to the protocol), but there is no indicator that could be used to make it.

Trusted Platform architectures may help to address this deficit if they enable prospective communication peers to gain insights about the configuration of machines at the far end. This allows statements and negotiations about the expteced behaviour of the technical systems and could enable explicit communication contracts prior to intended interactions between them. Operating system and application designs that build on a trusted core would be tailored to enforce such contracts are honoured for the lifetime of specific transactions.

This 'communication contract' model provides the equivalent of implicit contracts that precede direct human interactions. In essence, they state in a verifiable way that the participating technical systems are configured

- to disallow potentially harmful actions during the interval of the intended transaction if this is possible, and

– to quit the transaction, should a harmful action occur during this interval.

This model mirrors the idea of an implicit social contract and replicates it as an explicit contract at the technical level. For the duration of this contract or interaction, the participanting entities waive actions that are considered harmful for achieving a common goal. To some extent, this model could allow to separate the expectations that concern the behaviour of the IT system from those that concern the behaviour of their owner (the common view is that in computer mediated communication, the owner *is* the IT system. If we accept that

– advanced information technology is fundamentally differen from traditional communication tools because it can reify intentions of unknown and potentially hostile third parties in form of software agents,
– rogue agents software acting on the behalf of third parties can actively modify data and falsify the intentions of the originating party as a consequence,

we may not escape the conclusion that technical support is necessary to determine that advanced IT components behave, as expected, according to their configuration. A central aim and vision of Trusted Computing Platform technology is to support the decision whether IT technology is appropriate for and transparent with respect to the user's intentions.

1.4 Why Trusting Yourself May Not Be Good Enough

The fierceness of the debate that revolves around Trusted Computing Platforms indicates that a taboo has been touched. It is frequently coined as full control or maximized customer value; I suggest to consider the term *omnipotence*. It seems worthwhile to have a closer look at this attribute that is so strongly upheld. From a practical perspective, one might ask whether fits the current and future technical realities. From a more theoretical point of view, the compatibility of full control with the idea of communication contracts as outlined in the previous chapter is of some interest.

In practical terms, full control assumes full knowledge and understanding of what is actually happening on a system. Due to the ever increasing complexity of IT systems, however, this is becoming more illusory by the day. Systems are getting bigger, faster and more feature-rich. Most users have little if any idea about what is going on behind their graphical user interface. Even administrators frequently do not have a comprehensive understanding about what is actually happening on machines.

A quick and unrepresentative poll in my working environment suggests that the late 1980s were probably last time when a typical research scientist had a more or less comprehensive grasp of all layers of hardware and software on his desktop machine. The integrity of IT systems can be subverted at virtually all levels and stages of their operational cycle. It is hard to see how individuals could attest to the security properties of systems that are under their control without understanding them. In fact, they can't, which is why they rely on judgements of trusted third parties.

If owners typically cannot vouch for the integrity of their systems, we can only achieve the goal of better trustworthiness by enabling technical systems to vouch for their own status. This implies continuously monitoring and providing an audit log that indicates the system's trustworthiness to a remote peer.

What about the ueber-administrators and hackers, who really know the guts of their machines, work at the bleeding edge of technology and are the sources rather than receivers of vulnerability analysis and security patches? They may be able to make their system watertight; however, their system on its own can not communicate this in a trustworthy way. The system's trustworthiness must always be inferred from the reputation of its owner. This reputation only makes sense for another human, and has to be established out of band, if necessary by indirection via a trusted third party. Trusted Platform technology aims at providing a mechanism to communicate the actual system configuration in a trustworthy manner instead of relying on the credentials of its administrator.

1.5 On Interactions, Omnipotence, and Benevolence

Trust Yourself may be sufficient to solve your own problems in your own domain. In order to communicate and interact beyond its limits, you may have to convince remote peers that they can have confidence in elements of your domain.

This consideration is of some importance when we consider GRID-like scenarios or highly distributed electronic services. Here, we can not assume that services that run on our own behalf will run on platforms inside our own domain. Parts of our resources may work on behalf of others, e.g. because we have rented them out or because we have donated them for a distributed collaborative project. In order to manage their risk, other participants would want reasonable assurances that remote processes are not interfered with. On current systems, such assurances are hard to come by, as options for arbitrary interference tend to be a built in.

Today's systems tend to rely on the role of a benevolent superuser, a God on the machine, who can be in complete and unrestricted command of all system resources whenever it pleases her. For any given domain, its managers will encourage or enforce benevolence of superusers by means of incentives, policies, threats, separation of roles, monitoring etc.

In comparison, consumers of resources in external and remote domains find themselves at a huge disadvantage. They too have to take non-benevolent interference into account, however, without having means to estimate the risk, to monitor and to intervene. They can only fear that omnipotent roles on the IT equipment at the far end will not be abused; hope and faith have to make up for the lack of reliable indicators that this omnipotence is benevolent. Insisting on full control at all times perpetuates this asymmetry. On the other hand, waiving the option to execute full control during of an intended interaction can signal benevolent intent, break the asymmetry and enable interaction.

Waiving omnipotence might be the only way to unambiguously signal benevolence. In contractual terms: two parties agree to subject themselves to be-

havioural constraints in order to achieve a common goal. Trusted Computing Platforms aim to support this confidence on a technical level.

2 TCPA in a Nutshell

2.1 Motivation

The two major advantages of today's end systems and networks – openness and flexibility – are contributing factors to the current lack of confidence in IT security. Arguably, the extent to which a system should be open and flexible relies on the purpose it serves to its owner and his communicating peers at any given moment. In some situations, maximum openness and flexibility are desirable. In others, the exact opposite might be true, and we want the system's behaviour be constrained by a strict policy that exactly meets our expectations.

Policy-enforced behaviour of IT has been addressed by research in access control mechanisms, in particular for operating systems. Secure systems have traditionally been designed for environments where concerns of confidentiality, integrity and separation of roles are prevalent under almost all conditions, namely, for the military and financial sector.

Unfortunately, these designs tend to remove the advantages of openness and flexibility while requiring additional system management at the same time. Trusted Platforms attempt to combine the advantages of both worlds. They start from the understanding that in everyday situations, security is a flexible notion rather than an absolute goal: a system just has to be secure enough to be fit for purpose.

2.2 Approach

Trusted Platforms do not insist on security being provable under all conditions, even less so because the user may not understand and therefore not trust the proof. It is deemed more important that there a trusted party – which may be the user himself – vouches for the fact that a particular system configuration and policy is fit for a particular purpose. Focussing on trust and on vouching rather than on security as the core problem reflects a point that has already been mentioned: if IT systems are too complex, their users have no choice but to rely on attestations of third parties.

The TCPA efforts explicitly reflect a number of economical and legal constraints. Minimal price and small physical dimensions encourage to use the hardware component in a wide range of IT devices. In order to pre-empt problems with national export restrictions, the hardware does not support features such as bulk encryption. Consumer protection dictated designing TCPA as 'opt-in' technology that users can disable if they wish. It has been argued that power asymmetries between information providers and customers may coerce users into opting in. This could be a valid point, however, it addresses economic factors rather than technical features. At the technical level, TCPA provides temporary as well as permanent 'opt out'.

One of TCPA's foremost goals is to reliably record and report the configuration and state of a platform not only to the local user of a machine, but also peers he is communicating with by means of his computing equipment.

The trustworthiness of a system depends on its software configuration, its policy and its state. TCPA mechanisms can help to determine whether an expected configuration is factually present. Users can convince themselves that a system is good enough for an intended purpose either by trusting themselves or by basing their judgement on a third party that vouches for the system's 'fitness for purpose'. Note that 'fit for purpose' is a pragmatic notion could mean 'good enough' as well as 'secure'. Trusted Platforms support judgements about the level of risk that they might not behave as expected. Secure systems are designed with the goal to minimize or exclude risk. Clearly, secure systems can be built on top of Trusted Platform technology. Once a user accepts a configuration, the generic policy of a software system that makes use of Trusted Platform technology should be to monitor and to maintain this configuration for the duration of the intended user (trans)action. Systems that are built on top of TCPA technology will use its features to ensure the integrity of the system configuration once it has been accepted. This includes enforcement of any particular policy that is part of this configuration. How they do this, however, is not defined by TCPA.

2.3 Intentions

Given the breadth of TCPA's intended usage, security requirements vary widely due to different usage contexts and platforms. This can not be covered comprehensively by a technical specification. Therefore, TCPA only provides technology for some very generic, security related problems. The technology makes minimal assumptions about usage scenarios, assuming little more than every platform having an owner. It also reflects the common situation where users do not own the platforms they are working with.

Trusted Platforms technology as such is oblivious to specific policies, configurations, or software. This is the reason why there is no such thing as a TCPA-operating system or TCPA-certified applications. Operating systems and applications as well as any particular mechanism of policy decision and enforcement mechanisms lie squarely outside the scope of TCPA. Any attempt to directly equate TCPA with digital rights management, license enforcement or remote control is harmful inasmuch it diverts attention from the area where most, if not all of potential negative implications may arise. The threat is not from TCPA itself, but from concrete systems and software that are built *using* TCPA technology. Harmful legislation is another problem that can make matters worse.

2.4 Technology

The TCPA architecture consists of three major elements: hardware, software, and infrastructure. The interaction between these components is quite complex

and can only be outlined in this section, which refers to version 1.1b of the specification. Additional sources are mentioned at the end of this document.

Hardware: The hardware component (Trusted Platform Module or TPM) provides functionality that is roughly equivalent to that of a state of the art smartcard. It includes a random number generator, a generator for RSA key pairs, and a limited amount of non-volatile storage. As of March 2003, at least three vendors produce this hardware as a stand-alone chip. At the level of the chip's tamper-resistance, the non-volatile memory on the chip is considered protected from interference and prying. Like typical smartcards, a TPM chip is essentially a passive device in that it only performs actions at explicit request and does not include a general execution area. It never "takes charge", but relies on the main CPU.

Some of the non-volatile memory on the TPM is used to store two 2048 bit asymmetric key pairs. One of them, the Endorsement Key, is generated at the vendor's premises during production time and is the single unique identifier for the chip. The second, the Storage Root Key is generated when a customer takes ownership of the TPM. During this process, the prospective owner defines an authorization secret that he has to provide to the TPM from then on to enable it. The private components of both the Endorsement and the Storage Root keys are stored exclusively inside the TPM, i.e., they never become visible to software, the owner, or any other party. The owner can not use the private half of endorsement key to sign or encrypt data. In order to decrypt data that has been encoded using the public half of the endorsement key, knowledge of the authorization secret is required.

The remaining non-volatile memory on the TPM is organized as two sets of registers. A Platform Configuration Register (PCR) is designed to store values that represent the complete history of its modifications. It is not possible to set a PCR to arbitrary values; they can only be modified in a way that preserves the history of all its previous modifications. A Data Integrity Register (DIR) has the same size as a PCR. It can hold an arbitrary value that typically reflects the expected value of a corresponding PCR. Modifying Data Integrity Registers requires the knowledge of the TPM authorization secret and is therefore under the control of the TPM owner. Current TPMs have 16 PCRs and 16 DIRs.

Most TPM commands are combinations of the basic functions mentioned above: authorization secret, key protection, key generation, shielded configuration registers and integrity registers. Among other things, the TPM supports:

- employing asymmetric key pairs that can not be used by software, but only by a TPM,
- irrevocably transferring control over such key pairs from the TPM owner to other users of his system without the owner learning the private key component beforehand,
- logging system events in a non-reversible manner, supporting reliable auditing of the system's bootup and configuration,

- generating and using asymmetric key pairs whose private component will never be disclosed outside the TPM (non-migratable keys),
- binding the usage of TPM-generated keys to specific values of the configuration registers, thereby allowing to
- binding the capability to decrypt data to a specific platform state

Most of these operations are not provided by the TPM on its own, but need operating system and software support.

Software: TCPA compliant systems require two types of software. The first type, the Trusted platform Support Service or TSS, implements some complex functions that need multiple invocations of the TPM and symmetric encryption functionality. The second type, called 'Core Root of Trusted Measurement', is part of the platform firmware. It will typically reside in a BIOS or chipset and executed at an early stage of the platform bootup. Its task is to generate hash values of all binary code that is about to be executed and to log these values into the PCRs of the Trusted Platform Modules.

The core idea is to inductively extend this type of 'software measurement' from the firmware and BIOS to the operating system, OS services and applications. TCPA defines the chain of integrity verification up to the OS boot loaders. Specific boot loaders or operating systems are not covered by the specification. In this respect, TCPA is OS-neutral.

Infrastructure: TCPA based systems provide indicators that help to determine the level of confidence users can have in a given software environment. This judgement can be based on trusted statements of other parties. In order to communicate these statements, TCPA makes use of digital signatures, certificates, and public key infrastructure.

The first certificate concerns the unique identifier inside the TPM, the endorsement key. It attests that the private endorsement key resides on a TPM of a specific type, on this TPM alone and that it has never been disclosed to anyone (it could have been on the chip itself or in a secure environment of the chip manufacturer).. The second certificate attests that a specific TPM with a specific endorsement key has been properly integrated on a motherboard of a specific type. Platform credentials include a reference to a third kind of credential, the conformance certificate. It vouches for the fact that the combination of a TPM and a specified type of motherboard meet the TCPA specification, e.g., because both implement the Protection Profiles mentioned above.

The last certificate type can combine all aforementioned credentials in a single statement. The TCPA specifications envisages these 'identity certificates' to be issued as identifiers for Trusted Platforms. It is noteworthy that

- identity certificates do not need to reflect attributes of human users in any way, as they identify platforms;
- a single Trusted Platform can have an arbitrary number of identity certificates, hence multiple identities;

– requests for identity certificates do not require to prove platform ownership to a remote party. Proof of platform ownership has only be proven locally, as it is necessary to interpret the response to a certificate request.

2.5 Third Parties

As a matter of principle, all TCPA mechanisms *can* be used without involving external certificate authorities. Doing so, however, puts the burden of proof for the trustworthiness of all keys on the platform owners. They may decide to certify themselves, for example. In this case, the usefulness of these credentials will be limited to their immediate environment, i.e., their own policy domain, but that may be good enough.

Certificate Authorities (CAs) that issue TCPA identity certificates may follow arbitrary policies since the specification is agnostic about particular CA rules and platform configurations. In order to make the identity certificate useful beyond a particular domain at all, the Endorsement and Platform certificates will have to be verified. Additional requirements may concern the protection level attested by the conformance certificate. Identities can also be bound to specific configurations as represented by PCR values. In this case, a specific identity is supposed to only be usable if a platform has been booted in a defined manner and is in a defined state.

3 Reactions to TCPA

TCPA has met a lot of criticism from security experts, computer scientists and consumer protection organizations. TCPA critics object the technology on the grounds that it means DRM, less competition, less freedom – including freedom of choice – and less user control. Supporters of TCPA have upheld that much of the critique is based on speculation and limited understanding of the technology, and that mutual assurance for IT systems is a real and pressing issue that is independent of any given political and economic context which has to be addressed where it crops up: at the level of technology. Who is right?

The the ongoing controversy about TCPA can be characterized by two observations

– Many risks that have been mentioned refer to software architectures that could be built on top of TCPA, not to the hardware
– Many counter arguments are concerned with potential economic and political implications, not with technological aspects of TCPA

This is certainly true for Ross Anderson's TCPA/Palladium-FAQ, notwithstanding its merit in presenting a more than comprehensive list of potential risks. If for no other reasons, the world should be grateful to Anderson for having kicked off a necessary discussion. The TCPA debate has shown that there is ample demand for scenario planning in this area. This is likely to invigorate

further scientific research on economic, political and social implications of IT security.

These, however, seem to stem almost exclusively from specific system designs such as Palladium (which is now called Next Generation Secure Computing Platform or NGSCP) rather than from TCPA itself. Palladium is not TCPA, but adds to it a proprietary chipset, a modified CPU, and a proprietary operating system. This has since been acknowledged by both cyber activists and data protection agencies and is reflected in recent considerations.

What bones of contention are left from a cyber activist, privacy and consumer protection point of view? A document distributed by the German Chaos Computer Club (CCC) in in mid-March 2003 lists four points:

- assurances that no hidden channels exist on the TPM that leak the user's secret keys,
- transparency of certification mechanism,
- complete user control over all keys stored on his platform,
- the option to transfer keys between different computers.

This catalogue is remarkably similar to the one formulated by the conference of Federal and Regional data protection officers in Germany end of March 2003 (while this text was written):

- complete user control of their IT technology, access and modifications only as informed consent in every single case,
- transparency of available security functions,
- usage of software and access to documents without enabling third parties to create user profiles

Most of these points must be addressed as features of systems that are built on top of TCPA, as they are clearly outside the scope of TCPA technology itself. An exemption are the two last points mentioned by the CCC document:

- TCPA allows the platform owner to pass control of TPM-protected keys to others who are using the same platform. The owner has to do this explicitly , but once this has been done, it can not be reversed without the cooperation of the new owner. Demands to make this transfer reversible without the consent of the new owner would therefore break the TCPA philosophy.
- Transferring keys between platforms is possible as long as they have been marked as migratable. There are, however, keys that can not be migrated – conceptually or because they have been marked in this way by the owner. Conceptually, non-migratability applies e.g. to the endorsement key and to identity keys that are bound to the platform.

TCPA allows creating one's own identity certificates and supports to have an arbitrary number of them. It does not preclude CA policies that support pseudonymity. However, its very purpose is to prevent spoofing of these pseudonyms identities, i.e., transferred between different TCPA platforms. To

allow arbitrary migration of non-migratable objects between platforms would defeat the very purpose TCPA was designed for. This is only possible in cooperation between TPM owner and vendor and requires the source TPM to be discontinued as a result.

3.1 Trusted Platforms, Politics, and Law

By far the largest number of concerns have been voiced with respect to the introduction of TCPA based Digital Rights and License Management. Complacency of lawmakers and judges and legal muddle in this area could make it tempting for content providers to create a technological *fait accompli* that favour their own interests. They could unilaterally try to introduce specific software implementations without having a proper public debate beforehand, thereby cementing their own interpretation of copyright and fair use at the expense of general public benefit.

Due to the peculiarities of digital content, a compromise that satisfies all parties involved be hard to find. At this stage, we even seem to lack the appropriate vocabulary to talk about the issues involved. We seem to need radically different concepts of acquisition, ownership and transfer of the rights on this material. Clearly, we have to deal with the fact that digital content will increasingly reside in a distributed and networked IT infrastructure.

Whether DRM systems are based on hardware or not: their use should be endorsed by a proper political decision process. Unfortunately, political decision makers have not shown particular competence in the past when it came to finding the proper balance between interests of content producers, content distributors, consumers and the general public. Ignorance and apparent disinterest of many political decision makers plays into the of well organized lobbying groups.

Imbalance of power, e.g., between large commercial organizations and individual customers or public and private interest, is traditionally reflected in law such as on consumer protection or fair use. These laws restrict what can be regulated by private contracts.

TCPA supports building systems that allow enforcing behavioural control by software means of software and security architectures, and this type of enforcement can be made a prerequisite e.g. for commercial interactions. In this case, it could be interpreted as an invisible term of a contract. This can be considered a private affair between the interacting partners, or it can be be interpreted as an implicit element of contract formation or even law-making.

Will the effect of software that is necessary or even legally mandatory to participate in communication be framed in the context of contract law and free contract negotiation? Or will this be considered a matter of public policy because, in Lawrence Lessig's words, we have to acknowledge that this code is equivalent to law? I argue that much will depend on the answer of this question, and there are good reasons to prefer the second alternative. In democratic societies, it is a matter of principle the laws need to be legitimate. Legitimacy is ensured by free vote, public discussion and complete disclosure of every law and regulation that is passed.

As the current discussion about TCPA is in fact almost exclusively about software (i.e., operation system and application support), it revolves around system components that can be adapted to customer requirements and preferences. Software can be modified. Software components can reflect legal or contractual requirements. Software design and re-design decisions can be the result negotiations between providers, consumers, and lawmakers. Software can become a matter of public debate because its code can be inspected – if it is Open Code.

3.2 Trusted Platforms, Open Source, and Open Security

Recent legislation and court verdicts in the US have exposed major problems for code that is related to system security or Digital Rights management. This code is marked as non-inspectable and legally protected from reverse engineering. This makes it impossible for a non-privileged citizen to verify assurances by technical inspection without breaking the law. They are expected to accept some executables as a 'black box' and to take for granted the truthfulness of a declaration that these executables are protection mechanism. They are not entitled to check whether the implementation is fit for purpose or includes functionality that has not been mentioned.

Although reverse-engineering of content protection mechanisms might be outruled by law, it is still technically possible. Future version of CPUs and chipsets that make use of TCPA could change this drastically. Cryptographically secured communication between memory, IO, CPU and chipset could make the inspection of the executable code very hard, if not impossible even for IT specialists, let alone for an ordinary user.

It appears to be an interesting legal question whether users may be denied the right to inspect binaries during runtime, given that this software could potentially be a trojan horse. One could argue that this right of inspection could legitimately only denied for binaries whose source code has been fully disclosed and which have verifyably been generated from this disclosed source code by a trusted compiler. This would be a precondition for legitimately claiming a status of non-inspectability during runtime.

There are number of compelling questions about the potential impacts of Trusted Computing Platforms on Open Source software. For example: does TCPA lead to the proliferation of customer lock-in to proprietary solutions? If future TCPA based software severely impede consumers and increase transaction costs, then sheer lack of convenience might actually convince the customer to look for alternatives to proprietary solutions.

Even more puzzling questions have been raised with respect to the relevance of Open Code for Trusted Systems and the compatibility of Open Source licenses and the attestation of its security properties.

It has been claimed that such assurances can actually undermine the GPL, destroy Free Software, allow the GPL to be hijacked for commercial purposes and thereby demotivate idealistic programmers. The original argument put forward by Anderson is based on the notion of a "TCPA operating system" and assumptions that full use of TCPA features requires proprietary certificates, neither

of which is backed up by the specification. However, he has implicitely raised an important, more general point: does the attestation of security properties for Open Source software have implications for its status, flexibility, production process, and distribution?

A remarkable shake up has been caused by the US National Security Telecommunications and Information Systems Security Policy (NSTISSP). It requires that from July 1st, 2002 on, all government acquisitions of IT systems dealing with information security must be evaluated and validated according to the Common Criteria or equivalent. These compliance requirements also apply for any type of Open Source software. Oracle Corp.'s announcement to submit Red Hat Linux Advanced Server for a Common Criteria evaluation can be interpreted in this context. The distortion of the Open Source ecosystem and markets by information assurance has already started.

Evaluators will claim that security validation of Open Source adds value, although attestation can only ever refer to a particular version of the source code. If the code is altered, the attestation of the original code loses its validity. The attestation of security properties is typically external to the source code and therefore not subjected to the GPL. However, the validation of this very source code is only possible because it is is open to everyone in the first place. Flexibility as envisaged e.g. by the GPL seems to be at odds with assurances such as e.g. provided by a Common Criteria evaluation. The claim of adding value is therefore problematic. Two types of values are affected, and one of them seems to be increased at the expense of the other.

This is quite unfortunate because there are clear benefits of an Open Source approach to security in general and TCPA in particular. Microsoft's announcement to make the source code of its nexus 'widely available for review' indicates that there is a huge problem at the core of Trusted Computing: who guards the guardians? How can one be sure that trusted software components are trustworthy indeed and not trojan horses instead?

'Openness' suggests itself as a necessary element to give a convincing answer. It is not a sufficient one, as openness does not imply that software is more secure. Neither does TCPA make software more secure, whether proprietary or Open Source. TCPA based systems can support integrity verification of binaries before they are executed, and they can also support generating a reliable, non-forgeable audit trail. Whether or not a program is executed after the integrity check is beyond the scope of what TCPA can enforce, but has to be decided by the operating system.

4 Conclusions

Most arguments put forward against TCPA actually address points that are beyond its scope. They are potential problems of software and systems that are built on top of this technology. Some of them might not arise without TCPA, and one might object TCPA on these grounds. However, the dilemmas of *mutual* reassurance mentioned in the first part are real. It is hard to see how they could

be resolved without reliably reporting the system state, and it is even harder to see how this could be achieved without attaching a trustworthy hardware component to platforms.If they aren't resolved, we can not estimate the risk.

Attempts to combine TCPA and Open Source software are likely to require changes to design, implementation and quality assurance processes. Finding solutions to combine flexibility and security may well become a major creative challenge for the Open Source community. Traditional security certification procedures as those defined by the Common Criteria or ITSEC could prove to be unsuitable for this purpose. We may need new procedures, tools and certification models that can help to combine the flexibility of development and licensing with increasing demands for security assurances. Combined efforts of Open Source community, academia and industry seem to be necessary here. This looks like a promising field for further investigation.

5 Further Reading

The relation between trust and complexity has been been explored by Niklas Luhmann [1]. Helen Nissenbaum has applied some of his thoughts to online environments and pointed to the problem of full control as a trust inhibitor. [2]. As for the paradoxes of being God and a bread-eater at the same time, the reader may refer to the New Testament.

Motivation and approach of TCPA are explained in [3]. This book also gives an detailed account of TCPA's official intentions; as for the assumed unofficial ones, the reader can consult [4] and [5]. [3] also includes a comprehensive overview of TCPA's technical aspects, [6] and [7] have to be considered the ultimate reference on this, although a hard read. [8] is a readable analysis of someone who appears to have dug through the specification back-to-back. It seems to owe something to discussions on the cryptography and cypherpunks mailing lists that started on June 22, 2002. For critical reactions, just google the term TCPA. Rebuttals have been produced by the TCPA consortium [9] and Dave Safford [10]. Lawrence Lessig [11] has written about the politics of code, and Ross Anderson maintains an online repository about IT security economics [12].

Comments of the Chaos Computer Club [13] and the Conference of German Data Protection Officers [14] can be found online. This addressed Palladium/NGSCP only in passing, more details can be found online [15]. Readers may also want some additional information on some of the reasons why security validation for Open Software is in increasing demand [16].

References

[1] Niklas Luhmann: "Trust as a Reduction of Complexity", in: Trust and Power: Two Works of Niklas Luhmann, New York: John Wiley and Sons, 1979, pp. 24–31

[2] Helen Nissenbaum: "Can Trust Be Secured Online? A theoretical perspective", Etica and Politica, no. 2, December 1999.

[3] Siani Pearson: Trusted Computing Platforms. Upper Saddle River, NJ: Prentice Hall, 2003
[4] Ross Anderson: TCPA / Palladium Frequently Asked Questions (2002/03) http://www.cl.cam.ac.uk/ rja14/tcpa-faq.html
[5] Lucky Green: TCPA: the mother (board) of all Big Brothers (September 2002) http://www.cypherpunks.to/TCPA_DEFCON_10.pdf
[6] TCPA: TCPA Main Specification Version 1.1b (May 2002) http://www.trustedcomputing.org/tcpaasp4/specs.asp
[7] TCPA: TCPA PC Specific Implementation Specification Version 1.00 (September 2001) http://www.trustedcomputing.org/docs/TCPA_PCSpecificSpecification_v100.pdf
[8] Wintermute: "TCPA and Palladium Technical Analysis" (January 2003) http://wintermute.homelinux.org/miscelanea/TCPA%20Security.txt
[9] TCPA: TCPA Specification/TPM QA (October 2002) http://www.trustedcomputing.org/docs/TPM_QA_1016021.pdf
[10] Dave Safford: "Clarifying Misinformation on TCPA" (November 2002) http://www.research.ibm.com/gsal/tcpa/tcpa_rebuttal.pdf
[11] Lawrence Lessig: Code and other Laws in Cyberspace. New York: Basic Books (1999)
[12] Ross Anderson: "The Economics and Security Resource Page" http://www.cl.cam.ac.uk/ rja14/econsec.html
[13] Chaos Computer Club: "TCPA – Whom do we have to trust today"; (March 2003) https://www.ccc.de/digital-rights/forderungen
[14] Entschliessungen der 65. Konferenz der Datenschutzbeauftragten des Bundes und der Laender:
"TCPA darf nicht zur Aushebelung des Datenschutzes missbraucht werden" (March 2003) http://www.lfd.m-v.de/beschlue/entsch65.html
[15] Microsoft: Microsoft Next-Generation Secure Computing Base – Technical FAQ (February 2003)
http://www.microsoft.com/technet/treeview/default.asp?
url=/technet/security/news/NGSCB.asp
[16] Oracle Corp: "Oracle and Red Hat to submit Red Hat Linux Advanced Server for industry's first independent security evaluation of Linux",
http://www.oracle.com/corporate/press/1623351.html

On The Computation-Storage Trade-Offs of Hash Chain Traversal

Yaron Sella

The Hebrew University of Jerusalem,
School of Computer Science and Engineering,
Givat Ram, Jerusalem, Israel.
ysella@cs.huji.ac.il

Abstract. We study the problem of traversing a hash chain with a constant bound, m, on the number of hash-function evaluations allowed per each exposed link in the chain. We present a new, general protocol that solves this problem, and prove that its storage requirements are $k\sqrt[k]{n}$ chain links, where $k = m + 1$. We propose a new, natural criterion for evaluating the utility of a hash chain traversal protocol, which measures the length of the hash chain that the protocol traverses under fixed storage constraints. We present a new, specific protocol, tailored for the case $m = 1$, which improves the performance of the general protocol (with respect to the above criterion) by more than twice, and prove that the specific protocol is optimal in that sense.

Keywords: Amortization, hash chain traversal, optimality, pebbles

1 Introduction

Hash chains are a useful cryptographic primitive, that can provide efficient authentication services for a variety of applications. They have been proposed as a building block in electronic payment systems [1,10], in public-key one-time password (OTP) systems such as S/Key [4], in signing multicast streams [9], and in on-line auctions [11].

Recently, researchers started to investigate the cost of traversing hash chains. One can envision two extreme approaches for this problem: (1) Store only the seed of the hash chain, (2) Store the entire hash chain. Both methods are not very attractive. The first one has computational complexity $O(n)$, and the second has storage complexity $O(n)$ for a chain of size n. Influenced by the amortization techniques introduced by Itkis and Reyzin [5], Jakobsson [6] developed a method for traversing hash chains in computation $\lceil \log(n) \rceil$ and space $\lceil \log(n) \rceil + 1$. The technique was named "fractal hash chain traversal" due to the fractal storage and traversal patterns that it generates along the hash chain. The computational complexity has been further reduced by a factor of 2 in a recent work by Coppersmith and Jakobsson [2], at the price of a slightly more complex protocol than Jakobsson's. It is worth noting that some forward-secure signature schemes [5,7] use a sequence of activations of a one-way function (although not necessarily a

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 270–285, 2003.
© Springer-Verlag Berlin Heidelberg 2003

one-way *hash-function*) as a building block. Such schemes may also benefit from efficient techniques for hash chain traversal.

In this paper we consider applications that demand a constant upper bound, m, on the number of hash-function evaluations allowed per each link in the chain. Hash chain traversal protocols typically consist of rounds, where in each round a single (new) hash chain link is output, and some hash-function evaluations are performed in preparation for the next rounds. Our requirement is that the number of hash-function evaluations per round be bounded by a constant unrelated to n (not even to $\log(n)$ as in [6,2]). This seems like a very natural requirement, especially for applications with harsh timing constraints (e.g., real-time systems, heavily loaded servers). Of course, in return to enforcing a constant bound on the computation, such applications must accept some penalty in storage.

We present a general protocol that traverses a hash chain under this constraint, i.e., at most m hash-function evaluations are done per each link in the chain. The storage requirements of our general protocol are $k\sqrt[k]{n}$ links, where $k = m + 1$. The basic principle behind our protocol is that of a pipeline. We initialize the pipeline off-line by filling it with precomputed links. During on-line operation, we expose stored links from the pipeline, and at the same time keep refilling it by computing and storing new links. The entities responsible for the refill process are pebbles - dynamic helper points on a mission. A typical mission for a pebble is to traverse a certain section of the hash chain, and store specific links along the way. The protocol manages its pipelines in a recursive manner, where the recursion level depends on m. By setting $k = \log_2 n$, our general protocol becomes very similar to Jakobsson's protocol [6]. But one can also set k as a constant independent of n, thus allowing an application the flexibility to choose, where exactly to position itself on the time/space trade-off.

We then propose a new, natural criterion for evaluating the utility of hash chain traversal protocols. The idea is simply to fix the number of links that the protocol stores, and measure the length of the hash chain that the protocol can traverse. Of course, in this context, the size of the hash chain, n, is not a pre-determined, fixed constant, but rather a variable determined by the storage limitation and the utility of the protocol at hand. We call protocols that traverse the longest hash chain under some specific storage constraint *length-optimal*. Looking at the general protocol for the specific case $m = 1$ (i.e., at most one hash-function evaluation per each chain link), we see that it requires storage of $2\sqrt{n}$ links in order to traverse a hash chain of length n. In other words, under a storage limitation of ℓ links, the general protocol traverses a hash chain of size $\frac{\ell^2}{4}$ links. We present a specific protocol, tailored for the case $m = 1$, that under a storage limitation of ℓ links, traverses a hash chain of size $\frac{\ell(\ell+1)}{2}$ links. Obviously, with respect to the above criterion, the specific protocol is more efficient than the general protocol (for $m = 1$) by more than twice. We show that the specific protocol is the most efficient protocol with respect to the above criterion, by proving that it is length-optimal. Interestingly, both the general and the specific protocol are of fractal nature.

While considering relevant applications for efficiently traversed hash chains, we discovered a new hash chain construction, which is of independent interest. We call this new construction *double hash chain*. The novelty is that each link in a double hash chain is derived from a previous link using two hash-function operations. Double hash chains can be used for efficient signing of low bit-rate bit streams. An application in which this ability may be valuable is, for example, simultaneous contract signing.

The rest of this paper is organized as follows. Section 2 formally defines notations and terms used throughout the paper. Section 3 presents the general protocol, proves its correctness, and analyzes its storage requirements. Section 4 presents the specific protocol, shows that it is correct, analyzes its storage requirements, and proves its length-optimality. Double hash chains are introduced and discussed in Section 5.

2 Preliminaries

A *hash chain* is a sequence of values $< x_0, x_1, \ldots, x_n >$. The value x_0 is chosen at random. All other values are derived from it as follows: $x_i = f(x_{i-1})$, $(1 \leq i \leq n)$, where $f : \{0,1\}^* \rightarrow \{0,1\}^l$ is a hash-function such as SHA-1 [12]. A single value, x_i, is referred to as a *link* in the chain. The chain is generated from x_0 to x_n, and exposed from x_n to x_0. Hence, at different times, the natural roles of start and end of chain reverse. To avoid confusion, we use the following directions analogy: x_0 is placed on the left, and it is called the *leftmost* link. The chain is generated *rightward*, until we reach the *rightmost* link, x_n. The links of the chain are exposed *leftward* from x_n to x_0.

$$x_0 \xrightarrow{f()} x_1 \xrightarrow{f()} \ldots x_{n-1} \xrightarrow{f()} x_n \tag{1}$$

The hash chain defined above contains $n+1$ links, but we assume that at the outset, x_n is published as a public key. Hence, from here on, we refer to $X =< x_0, x_1, \ldots, x_{n-1} >$ as the full hash chain, and ignore x_n. Grouping one or more consecutive links of a hash chain constitute a *section* or a *sub-section*. The *size* of a hash chain section is the number of links in it. Thus, the size of the entire hash chain is n. Each section has a *leftmost* link and a *rightmost* link.

Our goal is to reveal links of the hash chain one-by-one from right to left, such that the number of hash-function evaluations per each exposed link is bounded by a constant. We denote this constant by $m \geq 0$, and we let $k = m + 1$. We assume that $n = b^k$ for some $b > 1$.

Definition 1. b-partition - *a b-partition of a hash chain section means dividing it into b sub-sections of equal size, and storing the leftmost link of each sub-section.*

Example: Let $X =< x_0, \ldots, x_{63} >$. Storing $\{x_0, x_{16}, x_{32}, x_{48}\}$ is a 4-partition of X.

Definition 2. Recursive b-partition - *a recursive b-partition of a hash chain section means applying b-partition recursively, starting with the entire section, and continuing with the rightmost sub-section, until the recursion encounters a sub-section of size one.*

Example: Let $X =< x_0, \ldots, x_{63} >$. Storing $\{x_0, x_{16}, x_{32}, x_{48}, x_{52}, x_{56}, x_{60}, x_{61}, x_{62}, x_{63}\}$, is a recursive 4-partition of X. Note that under the sectioning induced by recursive b-partition, a specific link can be the rightmost link of several sections, which were created at different levels of the recursion. For instance, the link x_{63} is the rightmost link of the sub-sections $x_{60} - x_{63}$ and $x_{48} - x_{63}$.

The definitions that follow use graph theory terminology pertaining to trees. As usual, a single, designated node is marked as the root of the tree. Father – children relationships are used to describe nodes connected by direct edges, where the father is closer to the root than its children. Leaves are nodes that have no children.

Definition 3. Recursive b-tree - *a recursive b-tree is a mapping of a recursive b-partition onto a tree as follows.*

- *Each node in the tree is a section or a sub-section. In particular, the entire hash chain is represented by the root of the tree.*
- *For every section L that is partitioned from left to right by sub-sections $L_1, \ldots, L_b$, the corresponding node N is the father of the corresponding nodes $N_1, \ldots, N_b$, and the children keep the same left to right order.*

Example: Figure 1 shows a recursive 3-tree with 4 levels. The root of the tree corresponds to the entire hash chain. Tree leaves which are most distant from the root, correspond to single links in the hash chain.

Definition 4. Neighborhood, neighbors - *sub-sections induced by recursive b-partition are neighbors if their corresponding nodes in the recursive b-tree have the same father. Depending on its position in the recursive b-partition, a sub-section can have a left neighbor, a right neighbor, or both.*

In order to traverse the hash chain with constant-bounded computation for each link exposed, our protocol uses *stored links* - links along the hash chain which are stored. We adopt the notion of pebbles from [5]. A *pebble* is a dynamic link that moves along the hash chain. Obviously, a pebble must start at a stored link, and must move from left to right. The mission of a pebble is to traverse a certain sub-section of the hash chain, and store some links along the way. When the mission is completed the pebble dies. As we will shortly see, the mission that pebbles get in the protocol that follows is to induce recursive b-partition.

3 General Protocol

This section presents a general protocol that traverses a hash chain of size n, while evaluating the hash-function at most m times per each exposed link. Recall

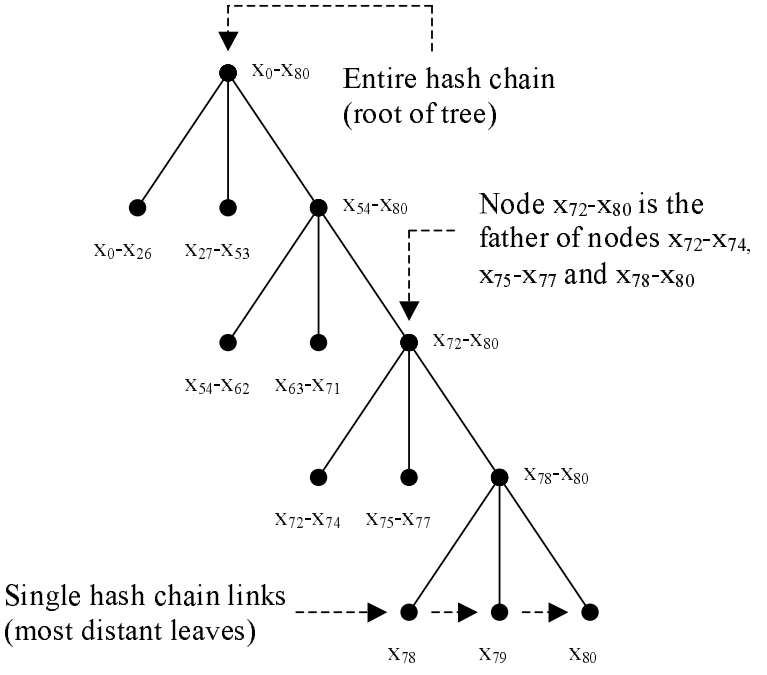

Fig. 1. A recursive 3-tree with 4 levels

that $k = m + 1$, and that $n = b^k$ for some $b > 1$. We start by describing the protocol, and continue by proving its correctness, and analyzing its storage requirements.

3.1 The Protocol

The protocol comprises two main phases: initialization and exposure loop. The initialization is done off-line, and the exposure loop is done on-line.

- **Initialization.** Perform a recursive b-partition on the entire hash chain. The recursion has k levels.
- **Exposure loop.**
  1. Expose and discard L, the current rightmost link in the hash chain.
  2. If $(L = x_0)$ stop.
  3. Identify S - the group of all sub-sections of size > 1 in the hash chain partitioning, for which L was the rightmost link.
  4. $\forall s \in S$: if s has a left neighbor $\widetilde{s}$, then place a pebble $\widetilde{p}$ at the leftmost link of $\widetilde{s}$. The mission of $\widetilde{p}$ is to perform recursive b-partition on $\widetilde{s}$, excluding the leftmost link of $\widetilde{s}$, which is already stored.
  5. Advance all pebbles one step rightward.
  6. Go to 1.

Example: Figures 2 and 3 demonstrate execution of the general protocol. Figure 2 shows it for the case $n = 9$, $m = 1$ ($\Rightarrow k = 2$, $b = 3$), i.e., a hash chain of $n = 9$ links, and a bound of $m = 1$ on the number of hash-function evaluations per each exposed link. Figure 3 shows it for the case $n = 27$, $m = 2$ ($\Rightarrow k = 3$, $b = 3$), i.e., a hash chain of $n = 27$ links, and a bound of $m = 2$ on the number of hash-function evaluations per each exposed link.

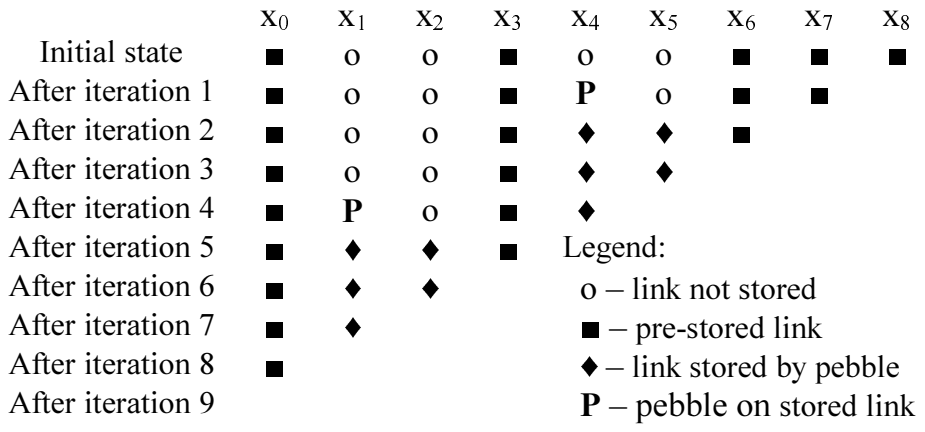

Fig. 2. Execution of the general protocol for $n = 9$, $k = 2$, $b = 3$

3.2 Correctness

In this section we prove two lemmas. Each lemma claims that our protocol satisfies a certain property. The first property is that every link that is exposed has been stored beforehand. The second property is that the protocol evaluates the hash-function at most $m = k - 1$ times per each exposed link. The two properties together imply that our protocol is correct.

Lemma 1. *Whenever the general protocol exposes the current rightmost link in the hash chain (step 1), that link has already been stored.*

Proof. By induction on the recursion level k.

Basis. For $k = 1$ the claim holds because the entire hash chain is pre-stored.

Step. We assume the claim is true for $k \leq h - 1$ and prove it for $k = h$. Let S be a hash chain that was processed by recursive b-partition with recursion level h. Denote the top level sub-sections of the hash chain as $s_1, \ldots, s_{b-1}, s_b$. The recursion level on sub-section s_b is $h - 1$. Therefore, by the induction hypothesis, the claim holds for s_b. As the rightmost link of s_b is exposed, a pebble starts to perform recursive b-partition on its left neighbor, s_{b-1}. The size of s_b and s_{b-1} is identical. Therefore, the pebble finishes its mission

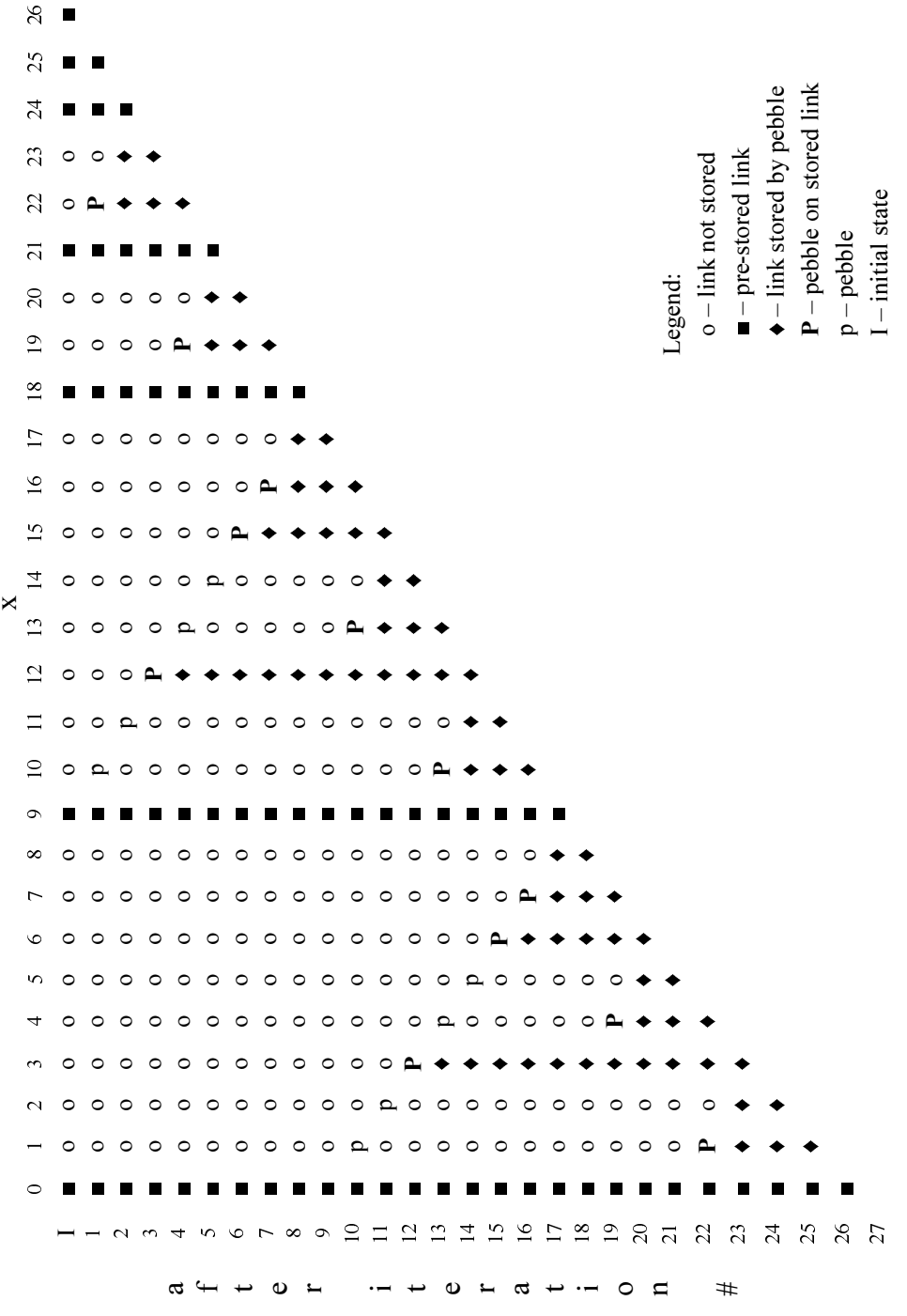

Fig. 3. Execution of the general protocol for $n = 27$, $k = 3$, $b = 3$

on s_{b-1} and dies, on the same iteration in which the leftmost link of s_b is exposed. A similar argument holds for the transition between s_{b-1} and s_{b-2}, and so on, until s_1.

Lemma 2. *The number of hash-function evaluations performed by the general protocol per each exposed link is at most* $m = k - 1$.

Proof. The protocol exposes one link in each iteration of the exposure loop. The number of hash-function evaluations that the protocol performs in each such iteration is equal to the number of live pebbles. Since $n = b^k$, there are exactly k levels in the recursive b-tree corresponding to the initial recursive b-partition. However, pebbles shall be created only for the $m = k - 1$ low levels, because the top level, which includes the entire hash chain, has no left neighbor. Indeed, in the very first iteration, right after x_{n-1} is exposed, exactly $m = k - 1$ pebbles are created.

We conclude the proof by showing that in every level of the recursive b-tree, at any given time, there can only be one live pebble. Consider recursion-level h. Let s_i be the sub-section at recursion-level h from which links are currently being exposed. Let s_{i-1} be s_i's left neighbor, and let s_{i-2} be s_{i-1}'s left neighbor. A pebble at recursion-level h must be in s_{i-1}. This pebble dies when the leftmost link of s_i is exposed. A new pebble is placed in s_{i-2} only when the rightmost link of s_{i-1} is exposed. Hence, before a new pebble is born at recursion-level h, the existing pebble must die, and the lemma follows.

3.3 Storage

We now consider the number of links that our protocol needs to store, and show that it is bounded by $k\sqrt[k]{n}$. At initialization, recursive b-partition is performed on the entire hash chain with recursion level k. As a result:

- At the top level $\sqrt[k]{n}$ links are stored.
- At all other $k - 1$ levels, $\sqrt[k]{n} - 1$ links are stored (because the leftmost link of each level is already stored by the level above it).

The total is $\sqrt[k]{n} + (k-1)(\sqrt[k]{n} - 1) = k\sqrt[k]{n} - (k-1)$ links. During the first iteration of the protocol, $k - 1$ pebbles are created. Since these pebbles advance to non-stored links, additional $k - 1$ links need to be stored. In summary, at this point, the number of stored links is bounded by $k\sqrt[k]{n}$. Our goal is to prove that this is also the upper bound on the protocol's storage requirements. We already saw in the proof of Lemma 2, that the number of pebbles is bounded by $k - 1$. It remains to bound the number of stored links in the different recursion levels of the recursive b-partition.

At the top level, $\sqrt[k]{n}$ links are stored when the protocol starts. As the protocol progresses, this number can only decrease. At the lower $k - 1$ levels, links are discarded, but the pebbles store new links. In the following we prove that for the lower $k - 1$ recursion levels, the number of stored links per level is bounded by

$\sqrt[k]{n} - 1$. In this proof, we will use k' as a *reverse-counter* for the recursion levels as follows: the lowest level is referenced by $k' = 1$, the top level is referenced by $k' = k$. The only exception is when $k = 1$, in which case we say that by convention $k' = 0$. Thus, k' counts the number of lower recursion levels. We refer to a specific level using the notation *at level* $k' = i$.

Lemma 3. *The number of stored links required per each of the lower* $k - 1$ *recursion levels of the recursive b-partition is at most* $\sqrt[k]{n} - 1$.

Proof. By induction on k'.

Basis. When $k' = 0$ (i.e., $k = 1$), the claim holds because zero lower levels must have zero stored links.

Step. We assume the claim is true for $k' \leq h - 1$, and prove that it is correct at level $k' = h$. Let S_b be the rightmost section at level $k' = h$, let $s_1, \ldots, s_{b-1}, s_b$ be S_b's sub-sections, and let S_{b-1} be S_b's left neighbor. We do not need to account for stored links inside the sub-sections $s_1, \ldots, s_{b-1}, s_b$; these are covered by the induction hypothesis. The links that interest us are the ones associated with S_b, namely, the links that separate between s_1 and s_2, s_2 and s_3, ..., s_{b-1} and s_b. Let C count the stored separator links in S_b. Let C' count the separator links stored by the pebble scanning S_{b-1}. We need to show that $C + C' \leq \sqrt[k]{n} - 1$. We already saw that after the initial recursive b-partition, $C = \sqrt[k]{n} - 1$ and $C' = 0$, so the inequality holds. Now observe that:

- The size of S_b, S_{b-1} is identical.
- The separator links in S_b, S_{b-1} divide them into sub-sections of equal size.
- A pebble starts scanning S_{b-1} on the same iteration in which the rightmost link of S_b is exposed.
- Links in S_b are exposed and links in S_{b-1} are traversed by the pebble in exactly the same speed - one per iteration.

Note that links in S_b are being exposed leftward, while S_{b-1} is being scanned rightward. Nevertheless, it follows that when the pebble in S_{b-1} stores a link, a separator link that had been stored in S_b is exposed and discarded. So we have $C' = C' + 1, C = C - 1$, and the inequality keeps holding. A similar argument holds for the transition between S_{b-1} and S_{b-2}, and so on, and the lemma follows.

In summary, we obtained that $k\sqrt[k]{n}$ is an upper bound on the number of links stored by our protocol. It is interesting to note, that by setting $k = \log_2 n$, we create a variant of our general protocol, which is very similar to Jakobsson's protocol [6]. Indeed, in this case, the protocol performs at most $\log_2 n - 1$ hash-function evaluations per each exposed link, and the number of links that must be stored is bounded by

$$k\sqrt[k]{n} = (\log_2 n)(n^{\frac{1}{\log_2 n}}) = (\log_2 n)(n^{\log_n 2}) = 2\log_2 n. \tag{2}$$

4 Specific Protocol

We now focus our attention on the scenario $m = 1$, namely, the specific case in which the number of hash-function evaluations allowed per each exposed link is one. In this case, the general protocol requires storage of $2\sqrt{n}$ links in order to traverse a hash chain of length n. In other words, if we let the general protocol store ℓ links, it can span a hash chain of size $\frac{\ell^2}{4}$. In this section we investigate whether this size can be extended.

More formally, we define the *length-efficiency* of a hash chain traversal protocol as the ratio between the length of the traversed hash chain and the number of stored links. We call protocols that achieve the best length-efficiency possible *length-optimal*. Obviously, length-optimal protocols are very desirable. In the following we present a length-optimal protocol for the case $m = 1$. The new protocol spans a hash chain of size $\frac{\ell(\ell+1)}{2}$ for the price of ℓ stored links. We call it the *specific protocol*, because it is tailored specifically for the case $m = 1$. We start by presenting the new protocol, continue by showing that it is correct, and conclude by proving that it is length-optimal.

4.1 The Protocol

The protocol comprises two main phases: initialization and exposure loop. The initialization is done off-line, and the exposure loop is done on-line. In the description of the specific protocol below, we assume that $n = \frac{\ell(\ell+1)}{2}$ for some $\ell > 1$.

- **Initialization.**
  1. Set $i = 0$, $sk = \ell$.
  2. Store x_i.
  3. If $(sk = 1)$ terminate.
  4. Set $i = i + sk$, $sk = sk - 1$.
  5. Go to 2.

  It is easy to verify that exactly ℓ links are stored during the initialization. To simplify the presentation, we denote these links in their left to right order as $y_0, y_1, \ldots, y_{\ell-2}, y_{\ell-1}$. Note that $y_0 = x_0$ and $y_{\ell-1} = x_{n-1}$.
- **Exposure loop.**
  1. Set $i = \ell - 2$
  2. Expose and discard L, the current rightmost link in the hash chain.
  3. If $(L = x_0)$ stop.
  4. If $(i > 0)$ and there is no pebble, place one at y_i and set $i = i - 1$. The mission of the pebble is to store every link between y_i and y_{i+1} (not inclusive).
  5. If there is a pebble advance it one step rightward.
  6. Go to 2.

Example: Figure 4 illustrates execution of the specific protocol for a hash chain of $n = 10$ links and $\ell = 4$.

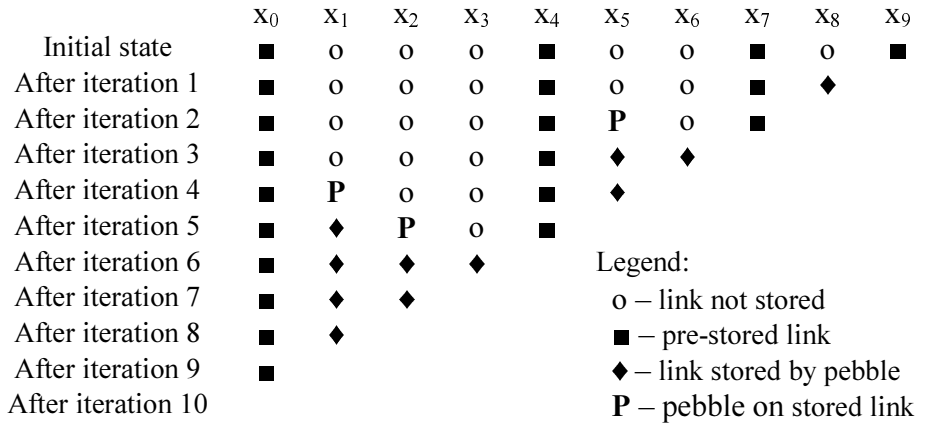

Fig. 4. Execution of the specific protocol for $n = 10$, $\ell = 4$

4.2 Correctness and Storage

It is easy to see that at initialization, the protocol stores exactly ℓ links out of the $n = \frac{\ell(\ell+1)}{2}$ links that make the entire hash chain. It is also easy to verify that during the exposure loop, there is at most one pebble active (in the last ℓ iterations there is no pebble at all). Therefore, whenever the pebble stores a link, the current rightmost link of the hash chain is discarded, so the number of stored links remains constant and equal to ℓ (until the last ℓ iterations, when it gradually decreases to zero). Finally, observe that $\forall i \geq 0$, the section that starts at y_{i+1}, contains exactly one link less than the section that starts at y_i, and that the pebble starts traversing y_i's section exactly when the rightmost link of y_{i+1}'s section is exposed. Since y_i itself is pre-stored, it follows that the pebble finishes preparing y_i's section just in time.

In summary, the specific protocol is correct because it evaluates the hash-function at most once per each exposed link, and the next link that needs to be exposed is always ready (i.e., pre-stored) on time. The storage requirements of the specific protocol are ℓ links for a hash chain of size $n = \frac{\ell(\ell+1)}{2}$ links.

4.3 Length-Optimality

In the following we prove that the specific protocol is length-optimal. The final theorem is preceded by four Lemmas upon which it is built.

Lemma 4. *Let P be a length-optimal hash chain traversal protocol for $m = 1$. Then P cannot discard a stored link before its exposure.*

Proof. Assume the contrary, i.e., that P discarded the link L_r on iteration i before L_r's exposure (notice that L_r cannot be the leftmost link in the hash chain). The exposure of L_r could not have been necessary in order to keep the storage limitations of P, because when $m = 1$ the number of stored links can

only decrease. There must be some iteration j in P that re-calculates L_r, such that between iterations i and j, L_r was not used. Notice that on iteration j, P must have the link L_{r-1} stored. Let P' be a new protocol derived from P as follows. P' does not discard L_r on iteration i. Clearly, the calculation of L_r by P' on iteration j becomes redundant. P' uses iteration j to calculate a new link, that is inserted between L_{r-1} and L_r. The resulting protocol P' keeps the constraint $m = 1$, uses the same amount of storage as P, and traverses a hash chain which is one link longer than P's. In contradiction to the length-optimality of P.

Lemma 5. *Let P be a length-optimal hash chain traversal protocol for $m = 1$, and let ℓ be the maximal number of links that P is allowed to store. Then P must pre-store upon initialization exactly ℓ links.*

Proof. Suppose P pre-stores less than ℓ links upon its initialization. Then one more link can be inserted on the left side of the hash chain, and P can be easily modified to traverse that additional link. In contradiction to the length-optimality of P.

Lemma 6. *Let P be a length-optimal hash chain traversal protocol for $m = 1$, and let ℓ be the maximal number of links that P is allowed to store. Then the number of iterations in which P does not need to evaluate the hash-function must be exactly ℓ.*

Proof. Based on Lemma 4, we know that re-calculating a link that had been stored contradicts length-optimality. So the only way for P to decrease the number of links that it stores is to perform iterations in which new links need not be calculated. Based on Lemma 5, we know that P must store ℓ links upon its initialization. By the time P terminates, the number of links that P must store must decrease to zero (note that storing a link that is never exposed immediately contradicts length-optimality). It follows that P must have at least ℓ iterations, in which the hash-function need not be evaluated. The number of such iterations cannot be greater than ℓ, because this implies that, at some point P stored more than ℓ links.

Lemma 7. *Let P be a length-optimal hash chain traversal protocol for $m = 1$ of the hash chain $< x_0, \ldots, x_{n-1} >$. Let y_k, y_{k+1} be two links pre-stored by P upon initialization, such that there are no pre-stored links between them. Let S_k be the section $[y_k, y_{k+1})$, and S_{k+1} be $[y_{k+1}, x_{n-1}]$. Then P can be transformed into an equivalent protocol, in which all the calculations (i.e., hash-function evaluation) in S_{k+1} occur before any calculation occurs in S_k.*

Proof. Consider the last calculation in S_k, after which there are still calculations in S_{k+1}. Suppose it occurs on iteration i, and let us mark it as HFE_i. Consider the first calculation after HFE_i which is in S_{k+1}. Suppose it occurs on iteration j ($j > i$), and let us mark it as HFE_j. The calculations HFE_i and HFE_j can safely exchange iterations due to the following.

1. Calculations in S_{k+1} cannot rely on the results of calculations in S_k since the pre-stored link y_{k+1} separates these two sections. According to Lemma 4, y_{k+1} cannot be discarded before its exposure, which clearly must happen after all the calculations in S_{k+1} are done.
2. A link in S_k cannot be exposed before a link in S_{k+1}, because S_k is located to the left of S_{k+1}. Therefore, the relevant link in S_k must be ready in time for exposure after the exchange.
3. Calculating a result earlier cannot affect its readiness. Therefore, the relevant link in S_{k+1} must be ready in time for exposure after the exchange.

By repeatedly exchanging calculations as described above, one eventually obtains a protocol that traverses the same hash chain, but with the additional property that all the calculations in S_{k+1} are done before any calculation in S_k.

Theorem 1. *Let P be a length-optimal hash chain traversal protocol for $m = 1$, and let ℓ be the maximal number of links that P is allowed to store. Then P traverses a hash chain of size $\ell(\ell+1)/2$.*

Proof. By induction on ℓ. For $\ell = 1$ the claim is trivial. We assume the claim holds for $\ell - 1$ (i.e., there exists a protocol that under storage limitation of $\ell - 1$ traverses a hash chain of size $(\ell - 1)\ell/2$), and prove that it holds for ℓ. Based on Lemma 5, we know that P must store ℓ links upon initialization. Based on Lemma 7, we know that P can be viewed as composition of two separate sub-protocols. The first one 'sees' the rightmost $\ell - 1$ pre-stored links of the hash chain. Clearly, it must be a length-optimal protocol for $\ell - 1$, otherwise the composition cannot be length-optimal. The second sub-protocol 'sees' only the leftmost link of the hash chain, which is, of course, pre-stored. Its contribution to the overall length of the traversed hash chain is the maximum number of links that it can calculate from that leftmost link. Based on Lemma 6, we know that the first sub-protocol gives the second sub-protocol $\ell-1$ iterations to evaluate the hash-function. Since the the combined protocol should also be length-optimal, the second sub-protocol must exploit all these iterations. As a result, in the transition point between the first and second sub-protocols, there are exactly ℓ stored links. Based on Lemma 6, if we want the combined protocol to be length-optimal, it cannot make any more computation, so the maximal number of links that the second sub-protocol can contribute is ℓ. The total is: $(\ell - 1)\ell/2 + \ell = \ell(\ell+1)/2$.

The length-optimality of the specific protocol follows directly from Theorem 1.

5 Double Hash Chains

In this section we present a real-world scenario, for which ultra-fast traversal of hash chains is beneficial, and propose a very efficient implementation suitable for this scenario based on double hash chains - a new type of hash chain. Double

hash chains are strongly related to Lamport's construction of one-time digital signatures from a one-way function [8]. The novelty of our approach is that we pack all the commitments in a hash chain. We believe that double hash chains are an interesting concept on their own, that may find uses in other applications as well.

Suppose a stream of bits needs to be communicated between two parties in an authenticated manner. Suppose further that the bits can be produced or consumed at a low rate, but they cannot be buffered due to application constraints. For example, Alice wants to buy bits from Bob, because Bob has an expensive device that produces truly random bits. Alice occasionally requests a small number of truly random bits from Bob, which she expands into a larger pseudo-random bit sequence. Since the bits are costly, Alice does not want to purchase many bits in advance, which would also force her to store and manage them. Alice prefers Bob to send her exactly what she needs for immediate use. Furthermore, Alice wants Bob to sign the bits, so that she can hold him responsible in case it turns out later that their quality was poor. Another example is simultaneous contract signing, during which two cautious parties prefer to sign the contract (or its hash) bit by bit. A third example is a sensor that periodically reports a state to a central alarm system. Because the sensor's state is of critical importance, the period must be quite short (e.g., one second), and consecutive reports cannot be aggregated. The report itself, however, may contain just a few bits (e.g., all is well, low suspicion, medium suspicion, high suspicion).

One possible solution is to number the bits, and sign each one with standard public-key signatures (e.g., RSA, DSS). Alternatively, one can amortize the price of a standard public-key signature across many bits, by establishing a series of one-time signature schemes, and using those for signing individual bits [3]. Note that only the first element in the series needs to be signed with a standard public-key signature. We propose a much more efficient solution based on a new type of hash chain.

5.1 Definition

Definition 5. Double Hash Chain - *a sequence of values* $< x_0, x_1, \ldots, x_n >$, *where* x_0 *is chosen at random, and* $x_i\,(i > 0)$ *is calculated as follows:*

$$x_i = f(str_1 \,||\, x_{i-1} \,||\, S^0_{i-1}) \,||\, f(str_2 \,||\, x_{i-1} \,||\, S^1_{i-1}) \tag{3}$$

The function $f()$ *is a one-way hash-function* $f : \{0,1\}^* \rightarrow \{0,1\}^l$ *such as SHA-1. The symbol* $||$ *denotes concatenation, and* str_1 *and* str_2 *are two different strings with the same length (e.g.,* str_1 = "a5a5a5a5", str_2 = "c3c3c3c3").

Example: The following formula illustrates how a double hash chain (dhc, for short) is constructed.

$$\begin{array}{ccccccc} & f(str_1 \,||\, x_0 \,||\, S^0_0) & & f(str_1 \,||\, x_1 \,||\, S^0_1) & & f(str_1 \,||\, x_{n-1} \,||\, S^0_{n-1}) & \\ x_0 & \longrightarrow & x_1 & \longrightarrow & \ldots x_{n-1} & \longrightarrow & x_n \\ & f(str_2 \,||\, x_0 \,||\, S^1_0) & & f(str_2 \,||\, x_1 \,||\, S^1_1) & & f(str_2 \,||\, x_{n-1} \,||\, S^1_{n-1}) & \end{array} \tag{4}$$

5.2 Usage

The function $f()$ is a global system parameter known by everyone, and so are the strings str_1 and str_2. For simplicity we assume two participants only - a signer and a verifier (the signer will also generate the dhc). Continuing with the scenario described in the beginning of this section, Bob is the signer and Alice plays the verifier.

Bob generates the dhc, and stores the arrays $S_0^0, \ldots, S_{n-1}^0$, $S_0^1, \ldots, S_{n-1}^1$ as his private key (in practice, they can all be generated from a single secret seed). Bob signs x_n using some public-key signature algorithm (e.g., RSA, DSS), and sends x_n to Alice as the dhc's public key. Alice accepts x_n if Bob's signature on it is correct. Bob initializes a counter i to $n-1$. Alice initializes a variable w to x_n.

An iteration in which Bob sends a signed bit, b, to Alice, and Alice verifies it proceeds as follows. Bob sends x_i and S_i^b to Alice, and decrements i. Let us denote these messages as x and S, respectively. Alice verifies that either the most significant half of w equals $f(str_1 \,||\, x \,||\, S)$, in which case $b = 0$, or the least significant half of w equals to $f(str_2 \,||\, x \,||\, S)$, in which case $b = 1$. Alice updates $w = x$. When the counter i maintained by Bob becomes negative, a new dhc must be established.

Note that efficient hash chain traversal techniques, such as the ones presented in [6,2] or in this paper, apply equally well to both regular and double hash chains.

5.3 Security

Due to space limitations, we do not include here a formal treatment of dhc's security goals, assumptions and proofs. However, a dhc is essentially a new type of hash chain. We therefore argue (informally) that as with ordinary hash chains, the security relies directly on the properties of the hash-function $f()$. The *preimage resistance* and *2nd preimage resistance* of $f()$ guarantee, that no one except Bob who generated the hash chain, can find its secrets, $S_0^0, \ldots, S_{n-1}^0$ and $S_0^1, \ldots, S_{n-1}^1$. These two properties also bind the two halves of each link together, preventing anyone (including Bob) from modifying a link (or part of it). The *collision resistance* of $f()$ guarantees that no one except Bob can sign. Collision resistance together with the fact that $str_1 \neq str_2$ prevent Bob from planting in the dhc rogue links, that allow him to repudiate authenticated bits committed by him. Of course, it is clear that a fundamental pre-requisite for security is to have enough bits in the relevant ingredients (x_i, S_i^0, S_i^1), such that birthday-paradox searches are infeasible.

Acknowledgments. I am grateful to Dahlia Malkhi and Ofer Margoninsky for their encouragement, helpful comments, and suggestions for improving the presentation of this paper.

References

[1] R. Anderson, C. Manifavas, and C. Sutherland. NetCard – A Practical Electronic Cash System. *Proc. of the fourth Cambridge Security Protocols Workshop*, pages 49–57, Cambridge, UK, 1996.

[2] D. Coppersmith, and M. Jakobsson. Almost Optimal Hash Sequence Traversal. *Proc. of The Fifth Conference on Financial Cryptography (FC'02)*, Bermuda, March 2002.

[3] R. Gennaro, and P. Rohatgi. How to sign digital streams. *Proc. of Crypto 97*, pages 180–197, 1997.

[4] N. Haller. The S/KEY one-time password system. *RFC 1760*, Internet Engineering Task Force, February 1995.

[5] G. Itkis, and L. Reyzin. Forward-Secure Signature with Optimal Signing and Verifying. *Proc. of Crypto 01*, pages 332–354, 2001.

[6] M. Jakobsson. Fractal Hash Sequence Representation and Traversal. *IEEE International Symposium on Information Theory (ISIT) 2002*, Lausanne, Switzerland, 2002.

[7] A. Kozlov, and L. Reyzin. Forward-Secure Signatures with Fast Key Update. *Proc. of The Third Conference on Security in Communication Networks (SCN'02)*, Amalfi, Italy, September 2002.

[8] L. Lamport. Constructing Digital Signatures from a One-way Function. *SRI International Technical Report SRI-CSL-98*, October 1979.

[9] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. *Proc. of IEEE Security and Privacy Symposium*, pages 56–73, May 2000.

[10] R.L. Rivest, and A. Shamir. PayWord and MicroMint: Two Simple Micropayment Schemes. *Proc. of The Fourth Cambridge Security Protocols Workshop*, pages 69–87, Cambridge, UK, 1996.

[11] S. Stubblebine, and P. Syverson. Fair On-line Auctions without Special Trusted Parties. *Proc. of The Fourth Conference on Financial Cryptography (FC'01)*, Grand Cayman, February 2001.

[12] FIPS PUB 180-1, Secure Hash Standard, SHA-1. www.itl.nist.gov/fipspubs/fip180-1.htm.

Verifiable Secret Sharing for General Access Structures, with Application to Fully Distributed Proxy Signatures*

Javier Herranz and Germán Sáez

Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
C. Jordi Girona, 1-3, Mòdul C3, Campus Nord, 08034-Barcelona, Spain
{jherranz,german}@mat.upc.es

Abstract. Secret sharing schemes are an essential part of distributed cryptographic systems. When dishonest participants are considered, then an appropriate tool are verifiable secret sharing schemes. Such schemes have been traditionally considered for a threshold scenario, in which all the participants play an equivalent role. In this work, we generalize some protocols dealing with verifiable secret sharing, in such a way that they run in a general distributed scenario for both the tolerated subsets of dishonest players and the subsets of honest players authorized to execute the different phases of the protocols.
As an application of these protocols, we propose a fully distributed proxy signature scheme. In this scheme, a distributed entity delegates its signing capability to a distributed proxy entity, which signs messages on behalf of the original one. We consider in both entities the aforementioned general distributed scenario.

1 Introduction

In a standard public key system, only the person who holds a secret key is able to perform the cryptographic task (signing or decrypting) corresponding to the related public key. Some situations, due to its importance, require the secret key of a system not to be held by a single person, but to be shared among a set of users. Participation of some authorized subset of users is necessary in order to perform the cryptographic task. In this way, both the security and the reliability of the system increase.

Secret sharing schemes are an essential component of these distributed cryptographic systems. In such schemes, shares of a secret value are distributed among the users, in such a way that only authorized subsets (those in the access structure) can recover the secret from their shares. When the system tolerates some users of the set to be dishonest, then the appropriate tool are verifiable secret sharing schemes. These schemes are constructed from secret sharing schemes

* This work was partially supported by Spanish *Ministerio de Ciencia y Tecnología* under project TIC 2000-1044.

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 286–302, 2003.
© Springer-Verlag Berlin Heidelberg 2003

by adding some public commitments that allow to detect users who try to falsify their shares.

Most of distributed cryptosystems (signature or decryption schemes) consider threshold access structures. That is, the cryptographic task could be performed only if a certain number of users of the group participate. For this reason, the most studied verifiable secret sharing schemes [3,12] are those that realize threshold access structures. However, there are situations in which the members of a group do not all have the same power or the same probability to be dishonest. In these cases, access structures more general than the threshold ones must be considered.

In this work we show that the standard threshold verifiable secret sharing schemes [3,12] can be adapted to run in a more general scenario. We also generalize some protocols that use as a basis these verifiable secret sharing schemes, such as the joint generation of shared discrete-log keys [6], and the distributed Schnorr signature scheme proposed in [20].

As an application of these protocols, we design a fully distributed proxy signature scheme, based on the proxy signature scheme proposed in [10]. Proxy signature schemes allow a potential signer A to delegate his signing capability to a proxy signer B (in some way, A tells B what kind of messages B can sign), and B signs a message on behalf of the original signer A. The receiver of the message verifies the signature of B and the delegation of A together.

A fully distributed proxy signature scheme can be applied in situations where a distributed entity wants to delegate some digital signing rights or capabilities to another one. For example, the central office of a bank can delegate to a branch office the capability to sign some kinds of documents (loans, mortgages) on behalf of the bank, whose secret key is distributed among members of the central office. The delegated capability will be distributed among members of the branch office. In real situations, these members do not all have the same power or influence within the office. Perhaps the policy of the bank is that a loan can be granted and signed by a branch office in the name of the bank only if the general manager and two other members of the branch office, or else any four members, agree. This is the motivation in order to consider structures that are more general than the threshold ones mainly considered until now.

Participation of an authorized subset of members of the first entity will be necessary to perform the delegation, and an authorized subset of the delegated entity will have to collaborate to generate a valid signature on behalf of the first entity.

Furthermore, the system must be able to tolerate the presence of some dishonest or inactive players within these distributed entities. Thus, an adversary who corrupts some of these tolerated subsets of players will not obtain enough information to break the system (for example, by forging a delegated signature).

Organization of the paper. In Section 2 we propose a framework for distributed protocols which is more general than the threshold one, and extend to this framework some threshold protocols such as verifiable secret sharing schemes, the joint generation of shared discrete-log keys and the distributed

Schnorr signature scheme. In Section 3, we review the area of proxy signature schemes, specifically the one of Lee, Kim and Kim [10], and explain the security requirements that these schemes must satisfy. In Section 4 we propose our fully distributed proxy signature scheme, which runs in the general framework introduced previously. Finally, in Section 5 we conclude by summing up our contribution and discussing some open problems.

2 Some Distributed Protocols in a General Framework

Distributed cryptographic protocols have two main advantages with respect to individual ones: an increase of the *security*, because more than one party must be corrupted by an attacker in order to obtain a secret key; and an increase of the *reliability*, because the protocol can be executed even if some parties are non-working at that moment for some reason. The idea is to share the secret information among the participants of the distributed entity.

In this section we deal with some distributed protocols related to the verifiable sharing of a secret value among a set of players. We will consider a framework which is more general than the threshold one. That is, those subsets of players authorized to perform some specific actions, such as the recovery of a secret or the signature of a message, as well as those subsets of dishonest players that the system is able to tolerate, will not be necessarily defined according to their cardinality.

We extend to this general framework the previous (threshold) proposals for verifiable secret sharing [12], joint generation of discrete-log keys [6] and threshold Schnorr's signature scheme [20].

2.1 Verifiable Secret Sharing

In a *secret sharing scheme*, a dealer distributes shares of a secret value among a set of players $\mathcal{P} = \{1, \ldots, n\}$ in such a way that only authorized subsets of players (those in the so-called *access structure*, denoted by $\Gamma \subset 2^{\mathcal{P}}$) can recover the secret value from their shares, whereas non-authorized subsets do not obtain any information about the secret (unconditional security). The structure Γ must be *monotone increasing*, that is, if $A_1 \in \Gamma$ and $A_1 \subset A_2$, then $A_2 \in \Gamma$.

Secret sharing schemes were introduced independently by Shamir [16] and Blakley [1] in 1979. Shamir proposed a well-known *threshold* scheme, in which the authorized subsets are those with more than t members (t is the threshold). Other works propose schemes realizing more general access structures; for example, *vector space secret sharing schemes* [2] are often used. An access structure Γ can be realized by such a scheme if, for some positive integer t and some vector space $E = K^t$ over a finite field K (in our context, it will be $K = \mathbb{Z}_q$ for some prime number q), there exists a function

$$\psi : \mathcal{P} \cup \{D\} \longrightarrow E$$

such that $A \in \Gamma$ if and only if the vector $\psi(D)$ can be expressed as a linear combination of the vectors in the set $\psi(A) = \{\psi(i) | i \in A\}$. If Γ can be defined in this way, we say that Γ is a *vector space access structure*; then we can construct a secret sharing scheme for Γ with set of secrets $\mathbb{Z}_q$: given a secret value $k \in \mathbb{Z}_q$, the dealer takes a random element $\mathbf{v} \in E = (\mathbb{Z}_q)^t$, such that $\mathbf{v} \cdot \psi(D) = k$. The share of a participant $i \in \mathcal{P}$ is $s_i = \mathbf{v} \cdot \psi(i) \in \mathbb{Z}_q$. Let A be an authorized subset, $A \in \Gamma$; then, $\psi(D) = \sum_{i \in A} c_i^A \psi(i)$, for some $c_i^A \in \mathbb{Z}_q$. In order to recover the secret, the players of A compute

$$\sum_{i \in A} c_i^A s_i \;=\; \sum_{i \in A} c_i^A \mathbf{v} \cdot \psi(i) \;=\; \mathbf{v} \cdot \sum_{i \in A} c_i^A \psi(i) \;=\; \mathbf{v} \cdot \psi(D) \;=\; k \mod q$$

Shamir threshold secret sharing scheme with threshold t is a particular case of vector space schemes, taking $\psi(D) = (1, 0, \ldots, 0)$ and $\psi(i) = (1, i, i^2, \ldots, i^{t-1})$.

Linear secret sharing schemes can be seen as vector space secret sharing schemes in which each player can have associated more than one vector. They were introduced by Simmons, Jackson and Martin [19], who proved that any access structure can be realized by a linear secret sharing scheme, although in general the construction they proposed results in an inefficient secret sharing scheme. These schemes have been considered under other names such as geometric secret sharing schemes or monotone span programs. In our work, we will consider any possible access structure, so we will know that there exists a linear secret sharing scheme realizing this structure. However, we will suppose for simplicity that this scheme is a vector space one.

A variation of these schemes are *verifiable secret sharing schemes*, which prevent the dealer and the players from cheating; each participant can check if his share is consistent with the shared secret. The two most used verifiable secret sharing schemes are the proposals of Pedersen [12] and Feldman [3]. Here we present a modification of the (threshold) verifiable secret sharing scheme proposed in [12]. We consider any access structure Γ. Furthermore, we must take into account which subsets of dishonest players can be tolerated by the system. Those subsets form the *adversary structure* $\mathcal{A} \subset 2^{\mathcal{P}}$, which must be *monotone decreasing*: if $B_1 \in \mathcal{A}$ is tolerated and $B_2 \subset B_1$, then $B_2 \in \mathcal{A}$ is also tolerated.

The situation is modelized by an *active adversary* who can corrupt, at the beginning of the protocol, all players of some subset $R \in \mathcal{A}$. During the execution of the protocol, the adversary controls the behavior of these players, deciding at each moment which players of R follow the protocol correctly and which ones lie, but the adversary cannot change the subset R in $\mathcal{A}$ that he has chosen at the beginning (we say that it is a *static* adversary). An obvious requirement is that the adversary cannot obtain the secret from the shares of the participants that he has corrupted, so the condition $\Gamma \cap \mathcal{A} = \emptyset$ must be satisfied.

In the threshold case, the structures $\Gamma = \{A \in 2^{\mathcal{P}} \;:\; |A| \geq t\}$ and $\mathcal{A} = \{B \in 2^{\mathcal{P}} \;:\; |B| < t\}$ have been usually considered. We are going to consider any possible structures Γ and $\mathcal{A}$ satisfying $\Gamma \cap \mathcal{A} = \emptyset$, and so we will use general linear secret sharing schemes (for simplicity, vector space ones) instead

of threshold secret sharing schemes. Here we present our generalization of the verifiable secret sharing scheme of Pedersen [12] to this general scenario.

Let p and q be large primes with $q|p-1$. Let g and h be generators of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. This will be the mathematical scenario in the rest of the paper. The set of players is $\mathcal{P} = \{1, \ldots, n\}$, and the access structure $\Gamma \subset 2^{\mathcal{P}}$ is defined by the function $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathbb{Z}_q)^t$. If the dealer wants to share the secret $k \in \mathbb{Z}_q$, in a verifiable way, he does the following:

1. Choose two random vectors in $(\mathbb{Z}_q)^t$:

$$\mathbf{v} = (v^{(1)}, \ldots, v^{(t)}) \quad , \quad \mathbf{w} = (w^{(1)}, \ldots, w^{(t)})$$

   such that $\mathbf{v} \cdot \psi(D) = k$.
2. Compute $(s_i, s_i') = (\mathbf{v} \cdot \psi(i), \mathbf{w} \cdot \psi(i)) \in (\mathbb{Z}_q)^2$ and send the pair (s_i, s_i') to player i, for $1 \leq i \leq n$.
3. Broadcast the public commitments $C_m = g^{v^{(m)}} h^{w^{(m)}} \in \mathbb{Z}_p^*$, for $1 \leq m \leq t$.

Each player i verifies that

$$g^{s_i} h^{s_i'} = \prod_{m=1}^{t} (C_m)^{\psi(i)^{(m)}} \tag{1}$$

where $\psi(i)^{(m)}$ denotes the m-th component of vector $\psi(i)$. If this equality does not hold, player i broadcasts a complaint against the dealer.

For each complaint from a player i, the dealer broadcasts the values $(s_i, s_i') = (\mathbf{v} \cdot \psi(i), \mathbf{w} \cdot \psi(i))$ satisfying equation (1). The dealer is rejected if he receives complaints from players of a subset that is not in the adversary structure $\mathcal{A}$, or if he answers a complaint with values that do not satisfy equation (1). Otherwise, the dealer is accepted.

This verifiable secret sharing scheme is computationally secure, assuming that the discrete logarithm problem in the group generated by g is hard (the proof is almost the same as that in [12] for the threshold case).

2.2 Joint Generation of Discrete-Log Keys

In this work, and roughly speaking, a distributed protocol is said to be *robust* if it always produces a correct output, even in the presence of some tolerated subset of dishonest players.

In [6] Gennaro, Jarecki, Krawczyk and Rabin use Pedersen's verifiable secret sharing scheme to design a protocol in which players in a set $\mathcal{P} = \{1, \ldots, n\}$ jointly generate a public key $y = g^x$ and shares of the corresponding secret key x, in such a way that t or more players can recover this secret key (threshold access structure). The idea is the following: each player i plays the role of a dealer and shares a random value k_i among the players. The secret key x will be the sum of some of these values.

We explain here the more general version considering any access structure $\Gamma \subset 2^{\mathcal{P}}$ (realizable, for simplicity, by a vector space scheme defined by a function ψ) and any adversary structure $\mathcal{A}$ satisfying some security and robustness conditions. If we want this protocol to be robust, we must make sure that, when we detect a dishonest subset of players in $\mathcal{A}$ and reject them from the protocol, an authorized subset in Γ still remains among the non-rejected players; this authorized subset of honest players can go on executing the protocol. That is, for any subset $R \in \mathcal{A}$, it must be $\mathcal{P} - R \in \Gamma$, or equivalently, $\mathcal{A}^c \subset \Gamma$, where $\mathcal{A}^c = \{\mathcal{P} - R \ : \ R \in \mathcal{A}\}$.

Combining this condition with the unforgeability condition $\Gamma \cap \mathcal{A} = \emptyset$, we have in particular that the structures $\mathcal{A}$ and Γ must satisfy the following condition: for all subset $R \in \mathcal{A}$ it is necessary $\mathcal{P} - R \notin \mathcal{A}$. We say that such a monotone decreasing structure $\mathcal{A}$ is $\mathcal{Q}^2$ in $\mathcal{P}$. Note that in the threshold case, this $\mathcal{Q}^2$ condition is equivalent to $n \geq 2t + 1$. These conditions were studied in [7] for the case of general multiparty computation.

The protocol for joint generation is as follows:

1. Each player i executes Pedersen's verifiable secret sharing scheme playing the role of a dealer. That is, he chooses two random vectors $\mathbf{v_i} = (v_i^{(1)}, \ldots, v_i^{(t)})$ and $\mathbf{w_i} = (w_i^{(1)}, \ldots, w_i^{(t)})$, in $(\mathbb{Z}_q)^t$, where $\mathbf{v_i} \cdot \psi(D) = k_i$ is the random secret distributed by player i, and sends to player j the pair $(s_{ij}, s'_{ij}) = (\mathbf{v_i} \cdot \psi(j), \mathbf{w_i} \cdot \psi(j))$, for $1 \leq j \leq n$. The public commitments are $C_{im} = g^{v_i^{(m)}} h^{w_i^{(m)}}$, for $1 \leq m \leq t$.
2. At step 1, players who cheat are detected and rejected. We define $F_0 = \{i |$ player i is not rejected at step 1$\}$. Since $\mathcal{A}^c \subset \Gamma$, we have that $F_0 \in \Gamma$. Furthermore, for all players $i \in F_0$ that pass this phase, there are valid shares s_{ij} corresponding to players j that form an authorized subset. Each player $j \in \mathcal{P}$ computes his share of the total secret as $x_j = \sum_{i \in F_0} s_{ij}$ (the total secret will be $x = \sum_{i \in F_0} k_i \ \in \mathbb{Z}_q$).
3. Now they want to compute the value $y = g^x = \prod_{i \in F_0} g^{k_i} \in \mathbb{Z}_p^*$. They use Feldman's verifiable secret sharing scheme (see [3] for the original threshold version):
   3.1. Each player $i \in F_0$ broadcasts $A_{im} = g^{v_i^{(m)}}$, for $1 \leq m \leq t$.
   3.2. Each player j verifies the values broadcast by all the other players in F_0. That is, for each $i \in F_0$, player j checks that

$$g^{s_{ij}} = \prod_{m=1}^{t} (A_{im})^{\psi(j)^{(m)}} \tag{2}$$

   If this verification is false, player j complains against i broadcasting the pair (s_{ij}, s'_{ij}) that satisfies verification at step 1 (Pedersen's scheme, equation (1) in Section 2.1), but does not satisfy equation (2).
   3.3. For players i who received some valid complaint at step 3.2, the other players j run the reconstruction phase of Pedersen's scheme to recover a vector $\tilde{\mathbf{v}}_\mathbf{i} = (\tilde{v}_i^{(1)}, \ldots, \tilde{v}_i^{(t)})$ such that $\tilde{\mathbf{v}}_\mathbf{i} \cdot \psi(j) = s_{ij}$, for all these

players j (depending on the case, they will recover exactly $\tilde{\mathbf{v}}_\mathbf{i} = \mathbf{v}_\mathbf{i}$, but this is not necessary). They can also recover the value k_i; this can be done because there are valid shares s_{ij} satisfying equation (1) at step 1 (Pedersen's scheme), corresponding to players j that form an authorized subset. All players in F_0 can compute, therefore, the correct value g^{k_i}. From the vector $\tilde{\mathbf{v}}_\mathbf{i}$, the correct commitment values $A_{im} = g^{\tilde{v}_i^{(m)}}$ can also be computed.

Then the public key $y = g^x$ can be obtained by any participant in the following way:

$$y = \prod_{i \in F_0} g^{k_i} = \prod_{i \in F_0} g^{\mathbf{v_i} \cdot \psi(D)} = \prod_{i \in F_0} \prod_{m=1}^{t} g^{v_i^{(m)} \psi(D)^{(m)}} = \prod_{i \in F_0} \prod_{m=1}^{t} (A_{im})^{\psi(D)^{(m)}}$$

After the execution of this protocol, we have the public key $y = g^x$, where $x = \sum_{i \in F_0} k_i$ is the corresponding secret key, and $x_j = \sum_{i \in F_0} s_{ij} = (\sum_{i \in F_0} \mathbf{v_i}) \cdot \psi(j) = \mathbf{v} \cdot \psi(j)$ is the share of player j corresponding to the secret x, where $\mathbf{v} = (v^{(1)}, \ldots, v^{(t)})$, with $v^{(m)} = \sum_{i \in F_0} v_i^{(m)}$. Besides, the final commitment values $A_m = g^{v^{(m)}}$ can be easily computed as $A_m = \prod_{i \in F_0} A_{im}$, for $1 \leq m \leq t$.

We note all these facts (parameters and outputs of the protocol) with the following expression:

$$(x_1, \ldots, x_n) \quad \overset{(\mathcal{P}, \Gamma, \mathcal{A})}{\longleftrightarrow} \quad ((x, y), \{A_m\}_{1 \leq m \leq t}, F_0)$$

The security and robustness of this protocol can be proved analogously to the proof in [6] (which corresponds to the threshold case $n \geq 2t + 1$).

2.3 Distributed Schnorr's Signature Scheme

In [14], Schnorr introduced the following signature scheme. As in the rest of the paper, let p and q be large primes with $q | p - 1$. Let g be a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. $H()$ denotes a collision resistant hash function.

A signer A has a private key $x_A \in \mathbb{Z}_q^*$ and the corresponding public key $y_A = g^{x_A}$. To sign a message M, A acts as follows:

1. choose a random $k \in \mathbb{Z}_q^*$
2. compute $r = g^k \mod p$ and $s = k + x_A H(M, r) \mod q$
3. define the signature on M to be the pair (r, s)

The validity of the signature is verified by the recipient by checking that $g^s = r y_A^{H(M,r)}$.

In [13], Pointcheval and Stern proved that, in the random oracle model, existential forgery under adaptively chosen message attack of Schnorr's scheme is equivalent to the discrete logarithm problem in the group generated by the element g.

Stinson and Strobl propose in [20] a threshold version of the Schnorr's signature scheme. The system can tolerate the presence of less than t dishonest players, whereas any subset of at least t honest players can compute a valid signature. They remark that the protocol can be adapted to run with other structures, using a general linear (verifiable) secret sharing scheme instead of the threshold secret sharing scheme (and its verifiable variants) of Shamir.

We present here an adaptation of their threshold scheme to the case of any access structure Γ and adversary structure $\mathcal{A}$, such that $\Gamma \cap \mathcal{A} = \emptyset$ and $\mathcal{A}^c \subset \Gamma$ (the justification for these combinatorial requirements is the same as in Section 2.2). We assume again, for simplicity, that Γ is a vector space access structure defined by a function ψ. The protocol has three parts.

Key generation: players in $\mathcal{P} = \{1, \ldots, n\}$ use the protocol explained in Section 2.2 to jointly generate shares of a secret key and the corresponding public key. The output will be:

$$(x_1, \ldots, x_n) \quad \overset{(\mathcal{P},\Gamma,\mathcal{A})}{\longleftrightarrow} \quad ((x, y), \{A_m\}_{1 \leq m \leq t}, F_0)$$

Signature generation: let H be a collision-free hash function, and M the message to be signed. If an authorized subset $F_1 \in \Gamma$, $F_1 \subset F_0$ wants to sign M, they do the following:

1. Players in F_1 run again the joint generation protocol of Section 2.2, with output
$$(k_1, \ldots, k_n) \quad \overset{(\mathcal{P},\Gamma,\mathcal{A})}{\longleftrightarrow} \quad ((k, r), \{C_m\}_{1 \leq m \leq f}, F_2)$$
where k is a random secret shared value in $\mathbb{Z}_q$, the value $r = g^k$ is public, and $F_2 \subset F_1$.
2. Each player $i \in F_2$ broadcasts
$$\gamma_i = k_i + x_i H(M, r)$$
3. Each player $j \in F_2$ verifies, for all $i \in F_2$, that
$$g^{\gamma_i} = \prod_{m=1}^{t} (C_m)^{\psi(i)^{(m)}} [(A_m)^{\psi(i)^{(m)}}]^{H(M,r)}$$
Define $F_3 = \{i |$ player i is not detected to be cheating at step 3$\}$.
4. Each player $i \in F_3$ computes $s = k + xH(M, r) \mod q$, in the following way: since $\mathcal{A}^c \subset \Gamma$, we have that $F_3 \in \Gamma$, so there exist public coefficients $\{\lambda_j^{F_3}\}_{j \in F_3}$ in $\mathbb{Z}_q$ such that $\sum_{j \in F_3} \lambda_j^{F_3} \psi(j) = \psi(D)$. Then, each player $i \in F_3$ computes
$$s = \sum_{j \in F_3} \lambda_j^{F_3} \gamma_j$$

The signature for the message M is the pair (r, s).

Verification: the verification phase is the same as in Schnorr's signature scheme; that is, the recipient cannot distinguish if the signature has been generated in a distributed way or not. The recipient checks that

$$g^s = ry^{H(M,r)}$$

We will use the expression

$$DistSchnSig(\mathcal{P}, \Gamma, \mathcal{A}, M, y, \{x_i\}_{i\in\mathcal{P}}, \{A_m\}_{1\leq m\leq f}) \;=\; (r, s)$$

to refer to an execution of the signature generation phase of this scheme, in which players of a set $\mathcal{P}$, with authorized subsets in the access structure Γ and tolerated subsets of dishonest players in the adversary structure $\mathcal{A}$, jointly generate a Schnorr's signature (r, s) on a message M, using the public key y, shares $(x_1, \ldots, x_n)$ of the secret key x, and commitment values $A_m = g^{v^{(m)}}$ for the components $v^{(m)}$ of the vector that in fact distributes the shares of x.

Following the security proof that appears in [20], this distributed signature scheme can be proved to be as secure as Schnorr's signature scheme. The idea of the proof is to prove that the protocol is *simulatable*; that is, given an adversary against the scheme, who can corrupt a subset of players of $\mathcal{A}$, there exists an algorithm which outputs values that are computationally indistinguishable from the values that the adversary views during a real execution of the protocol. Then, assuming that this adversary against the distributed scheme is successful in forging a signature under a chosen message attack, both this fact and the simulability of the distributed protocol can be used to construct an adversary against the original Schnorr's scheme, which is also successful in forging a signature under a chosen message attack. But in the random oracle model, this is equivalent to solving the discrete logarithm problem, as it was shown in [13]. We can therefore conclude that the distributed version of Schnorr's signature scheme presented in this section has the same level of security than the original scheme, in the random oracle model (see [20] for the details of the proof in the threshold case).

The protocol is also robust, if $\mathcal{A}^c \subset \Gamma$. This is due to the fact that there is always a subset in Γ that passes all the verification tests, and so players of this subset can finish the protocol correctly.

3 Proxy Signatures

As an application of the three distributed protocols explained in Section 2, we propose in this work a fully distributed proxy signature scheme that runs in a general scenario. Before giving the details of this new scheme in Section 4, we review in this section the basics of proxy signatures and the specific proxy signature scheme in which our distributed scheme is based.

The concept of proxy signature was introduced by Mambo, Usuda and Okamoto in [11]. They classified these signatures according to the delegation type and

the protection of the proxy signer. Kim *et al.* [9] included warrant information in these schemes; that is, the signer A sends to the proxy B a signed message in which A explicitly delegates its signing capability to B, allowing B to sign some kind of messages (specified in the warrant information) on behalf of A.

The idea of these proxy signature schemes is the following: A sends a message and its signature to a proxy signer, B, who uses this information to construct a proxy key, which B will use to sign messages on behalf of A. This proxy key must contain some authentic information about the proxy signer, if we want these schemes to satisfy the security requirements of proxy signatures listed in the work of Mambo *et al.* [11]:

(i) **Strong unforgeability:** only a designated proxy signer can create a valid proxy signature for the original signer (even the original signer cannot do it).
(ii) **Verifiability:** a verifier of a proxy signature will be convinced of the original signer's agreement on the signed message.
(iii) **Strong identifiability:** a proxy signature determines the identity of the corresponding proxy signer.
(iv) **Strong undeniability:** after creating a valid proxy signature for an original signer, the proxy signer cannot repudiate this signature against anyone.

In [10] Lee, Kim and Kim briefly modify the proposal of [9]: now the proxy signer B and the original signer A play asymmetric roles in the generation of a proxy signature, and so the warrant information must not contain an explicit delegation of A's signing capability. Besides, A does not need to designate a specific proxy signer. In [10], the authors add a new security requirement to proxy signature schemes (which their scheme, as well as that proposed in [9], satisfies):

(v) **Prevention of misuse:** the proxy signer cannot use the proxy key for other purposes than generating a valid proxy signature. That is, he cannot sign, with the proxy key, messages that have not been authorized by the original signer.

3.1 The Proposal of Lee, Kim, and Kim

The following proxy signature scheme was introduced in [10]. It is based on the proposal of Kim *et al.* [9], with the difference that the warrant information signed by the original signer must not explicitly include either his identity or the identity of the proxy signer. This is possible because the original signer and the proxy signer do not play the same role in the generation of a proxy signature, and so the verifier can identify both of them.

Original signer A has the key pair (x_A, y_A), with $y_A = g^{x_A}$, whereas the (future) proxy signer B also has his user key pair (x_B, y_B), with $y_B = g^{x_B}$.

Generation of the proxy key: the original signer A uses Schnorr's scheme to sign warrant information M_ω, which should specify which messages A will allow the proxy to sign on his behalf, the validity dates of the delegation, etc.

That is, A chooses at random $k_A \in \mathbb{Z}_q^*$, and computes $r_A = g^{k_A}$ and $s_A = k_A + x_A H(M_\omega, r_A) \mod q$. Signer A sends (M_ω, r_A, s_A) to a proxy signer B secretly (in fact, only the value s_A must remain secret, the values M_ω and r_A should be broadcast). Then B verifies the validity of the Schnorr's signature:

$$g^{s_A} = r_A y_A^{H(M_\omega, r_A)}$$

If the verification is correct, B computes his proxy key pair (x_P, y_P) as

$$x_P = x_B + s_A \ , \ y_P = g^{x_P} \ (= y_B r_A y_A^{H(M_\omega, r_A)})$$

Proxy signature generation: in order to create a proxy signature on a message M conforming to the warrant information M_ω, proxy signer B uses Schnorr's signature scheme with keys (x_P, y_P) and obtains a signature (r_P, s_P) for the message M. The valid proxy signature will be the tuple

$$(M, r_P, s_P, M_\omega, r_A)$$

Verification: a recipient can verify the validity of the proxy signature by checking that M conforms to M_ω and the verification equality of Schnorr's signature scheme with public key $y_A^{H(M_\omega, r_A)} r_A y_B \ (= y_P)$; that is

$$g^{s_P} = r_P (y_B r_A y_A^{H(M_\omega, r_A)})^{H(M, r_P)}$$

This proxy signature scheme can be proved to be as secure as Schnorr's signature scheme. Note also that other signature schemes can be used instead of Schnorr's one, provided that these schemes use keys of the form (x, y), with $y = g^x$; ElGamal signature scheme and DSS verify this condition.

4 Fully Distributed Proxy Signatures

In this section, we propose a distributed proxy signature scheme based on the proxy signature scheme of Lee *et al.* [10].

There are various proposals of distributed (threshold) proxy signature schemes. Zhang's proposal [21] is not strongly unforgeable, because the original signer can impersonate the proxy signer. Kim *et al.* [9] also proposed a threshold version of their proxy signature scheme. Hwang, Lin and Lu [8] adapt the threshold scheme of Kim *et al.* to the case in which the verifier of the proxy signature must be able to identify which specific players in the proxy entity have signed the message. All these schemes distribute only the power of the proxy signer that signs messages on behalf of the original signer. Why not also distribute the original signer, and in this way increase the security and reliability also in the delegation phase of the scheme?

Our proxy signature scheme is the first that is fully distributed, in the sense that we distribute both the original and the proxy signer. We consider general structures for the authorized subsets and for the tolerated subsets of dishonest

players. Finally, our scheme is based on the proxy signature scheme of Lee *et al.* [10], and so the original signer entity does not need to include explicitly his identity, nor the identity of the proxy signer in the warrant information that it signs.

4.1 The Scenario

We must think of entities A and B as sets of players $A = \{P_1, \ldots, P_{n_A}\}$ and $B = \{Q_1, \ldots, Q_{n_B}\}$. We consider general monotone increasing access structures $\Gamma_A \subset 2^A$ and $\Gamma_B \subset 2^B$ in these sets. Furthermore, the system will tolerate the presence of some coalitions of dishonest players, those in the adversary structures $\mathcal{A}_A \subset 2^A$ and $\mathcal{A}_B \subset 2^B$, which must be monotone decreasing. The scheme will be unforgeable even if some players in A and some players in B are corrupted and exchange their secret information, provided $\Gamma_A \cap \mathcal{A}_A = \emptyset$ and $\Gamma_B \cap \mathcal{A}_B = \emptyset$, of course. Finally, we require $\mathcal{A}_A^c \subset \Gamma_A$ and $\mathcal{A}_B^c \subset \Gamma_B$, in order to give robustness to the scheme, in the same way as in Sections 2.2 and 2.3.

We assume, for simplicity, that there exists a function $\psi_A : \{D\} \cup A \longrightarrow (\mathbb{Z}_q)^{t_A}$, for some positive integer t_A, such that a subset $J_A \subset A$ is in Γ_A if and only if $\psi_A(D) \in \langle \psi_A(j) \rangle_{P_j \in J_A}$, and the same for the structure Γ_B with a certain positive integer t_B and a certain function ψ_B.

Any subset of A whose honest players form a subset in Γ_A can delegate A's signing capability, and any subset of B whose honest players form a subset in Γ_B can sign a message on behalf of entity A.

4.2 Our Proposal

The protocol that we present has four parts:

Generation of the Entities' Keys

Players in A jointly generate a public key and shares of the corresponding secret key, using the protocol in Section 2.2. Players in B do the same. The result is:

$$(x_{A,1}, \ldots, x_{A,n_A}) \overset{(A,\Gamma_A,\mathcal{A}_A)}{\longleftrightarrow} ((x_A, y_A), \{A_m\}_{1 \le m \le t_A}, F_{0,A})$$

$$(x_{B,1}, \ldots, x_{B,n_B}) \overset{(B,\Gamma_B,\mathcal{A}_B)}{\longleftrightarrow} ((x_B, y_B), \{B_\ell\}_{1 \le \ell \le t_B}, F_{0,B})$$

Distributed Generation of the Proxy Key

In this phase, players in entity A sign a warrant information M_{ω_A}, using the first part of the distributed Schnorr's signature scheme explained in Section 2.3. However, they do not obtain the explicit signature, but shares of it (thus preventing the possibility of one dishonest participant in A sending this secret signature to a dishonest participant in entity B). Then they send some information to players in entity B. Each player in B then computes, from this

information, his share of the proxy key, which will later be used to generate a proxy signature in a distributed way. This subprotocol is as follows.

1. Players in A execute the first step in the signature generation phase of the distributed Schnorr's signature scheme explained in Section 2.3. That is, they run the joint generation protocol of Section 2.2, with output

$$(k_{A,1},\ldots,k_{A,n_A}) \quad \overset{(A,\Gamma_A,\mathcal{A}_A)}{\longleftrightarrow} \quad ((k_A,r_A),\{C_m\}_{1\leq m\leq t_A},F_{1,A})$$

   The values $r_A = g^{k_A}$ and M_{ω_A} are made public.
2. Each player $P_i \in F_{1,A}$ computes

$$\gamma_i = k_{A,i} + x_{A,i} H(M_{\omega_A}, r_A) \bmod q$$

   as his share of the value $s_A = k_A + x_A H(M_{\omega_A}, r_A) \bmod q$.
3. Each player $P_i \in F_{1,A}$ distributes the value γ_i, verifiably among the players in entity B, in such a way that any subset in Γ_B can recover this value. He uses Feldman's scheme [3]; that is, P_i chooses a random vector $\mathbf{v_i} = (v_i^{(1)},\ldots,v_i^{(t_B)})$ in $\mathbb{Z}_q^{t_B}$ such that $\mathbf{v_i}\cdot\psi_B(D) = \gamma_i$, he makes public the commitment values $D_{i\ell} = g^{v_i^{(\ell)}}$, for $1 \leq \ell \leq t_B$, and sends to each player $Q_j \in B$ the share $s_{ij} = \mathbf{v_i}\cdot\psi_B(Q_j)$.
4. The correct commitments $\{A_m\}_{1\leq m\leq t_A}$ and $\{C_m\}_{1\leq m\leq t_A}$ corresponding to the sharing of the secret values x_A and k_A, respectively, must be in some way publicly revealed to all players in entity B. For example, each player $P_i \in F_{1,A}$ sends these commitments to all players in B, who can distinguish which commitments are the correct ones because they are sent by a subset of $F_{1,A}$ that is not in $\mathcal{A}_A$.
5. Each player $Q_j \in B$ checks, for any received share s_{ij}, that

$$\prod_{\ell=1}^{t_B} (D_{i\ell})^{\psi_B(D)^{(\ell)}} = \prod_{m=1}^{t_A} (C_m)^{\psi_A(P_i)^{(m)}} [(A_m)^{\psi_A(P_i)^{(m)}}]^{H(M_{\omega_A},r_A)}$$

   and that

$$g^{s_{ij}} = \prod_{\ell=1}^{t_B} (D_{i\ell})^{\psi_B(Q_j)^{(\ell)}}$$

   If either of these two checks fails, Q_j broadcast a complaint against P_i. If P_i receives complaints from players that form a subset of B that is not in $\mathcal{A}_B$, then he is rejected. Let $F_{2,A}$ be the subset of players in A that pass this verification phase. Since $\mathcal{A}_A^c \subset \Gamma_A$, we have that $F_{2,A} \in \Gamma_A$.
6. Players of B publicly fix coefficients $\{\lambda_i^{F_{2,A}}\}_{P_i\in F_{2,A}}$ in $\mathbb{Z}_q$ such that $\psi_A(D) = \sum_{P_i\in F_{2,A}} \lambda_i^{F_{2,A}} \psi_A(P_i)$. Then the equality $\sum_{P_i\in F_{2,A}} \lambda_i^{F_{2,A}} \gamma_i = s_A$ holds, and each player $Q_j \in B$ uses these fixed coefficients to compute his share of the value s_A as

$$s_{A,j} = \sum_{P_i\in F_{2,A}} \lambda_i^{F_{2,A}} s_{ij} \bmod q \ .$$

In effect, if $J_B \in \Gamma_B$, there exists coefficients $\{\lambda_j^{J_B}\}_{Q_j \in J_B}$ in $\mathbb{Z}_q$ such that $\psi_B(D) = \sum_{Q_j \in J_B} \lambda_j^{J_B} \psi_B(Q_j) \mod q$. Then it is not difficult to see that $\sum_{Q_j \in J_B} \lambda_j^{J_B} s_{A,j} = s_A \mod q$, and that $\{s_{A,j}\}_{Q_j \in B}$ is a perfect sharing of the secret s_A, according to the access structure Γ_B.

7. Each player $Q_j \in B$ computes $x_{P,j} = x_{B,j} + s_{A,j} \mod q$ as his share of the secret proxy key $x_P = x_B + s_A \mod q$. The public proxy key is computed as $y_P = g^{x_P} = y_B r_A y_A^{H(M_{\omega_A}, r_A)} \mod p$.

Note that the vector that in fact shares the secret value s_A among the participants of B is

$$\mathbf{v} = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} \mathbf{v_i} = (v^{(1)}, \ldots, v^{(t_B)}) \ ,$$

where $v^{(\ell)} = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} v_i^{(\ell)}$, for $1 \leq \ell \leq t_B$. Therefore, the commitment values V_ℓ corresponding to the components $v^{(\ell)}$ of this vector $\mathbf{v}$ can be publicly computed from the commitments $D_{i\ell}$ of the components $v_i^{(\ell)}$ of the vectors $\mathbf{v_i}$, for $P_i \in F_{2,A}$ as follows:

$$V_\ell = g^{v^{(\ell)}} = g^{\sum\limits_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} v_i^{(\ell)}} = \prod_{P_i \in F_{2,A}} (g^{v_i^{(\ell)}})^{\lambda_i^{F_{2,A}}} = \prod_{P_i \in F_{2,A}} (D_{i\ell})^{\lambda_i^{F_{2,A}}}$$

Finally, the commitments corresponding to the components of the vector that shares the secret proxy key $x_P = x_B + s_A \mod q$ will be $U_\ell = B_\ell V_\ell$, for $1 \leq \ell \leq t_B$.

Note also that another possible strategy is to have an authority that receives the shares γ_i from players in A, computes the secret value s_A from these shares, and redistributes shares of s_A among players in B. This solution reduces the total number of communications of the scheme, but it has some drawbacks: the authority must be fully trusted and reliable (opposite to the philosophy of this work), and a bottleneck in the system is possible.

Distributed Generation of a Proxy Signature

If the players of entity B want to sign a message M conforming to M_{ω_A} on behalf of entity A, they execute

$$DistSchnSig(B, \Gamma_B, \mathcal{A}_B, M, y_P, \{x_{P,j}\}_{j \in B}, \{U_\ell\}_{1 \leq \ell \leq t_B}) \ = \ (r_P, s_P)$$

The proxy signature is the tuple $(M, r_P, s_P, M_{\omega_A}, r_A)$.

Verification

The recipient of a proxy signature can verify its validity by checking that

$$g^{s_P} = r_P (y_B r_A y_A^{H(M_{\omega_A}, r_A)})^{H(M, r_P)}$$

4.3 Security and Robustness of the Scheme

Our fully distributed proxy signature scheme can be proved to achieve the same level of security achieved by the proxy signature scheme of Lee *et al.*, in our case against an adversary who can corrupt a subset of players $R_A \in \mathcal{A}_A$ and a subset $R_B \in \mathcal{A}_B$ at the same time. Due to lack of space, we do not give formal proofs in this paper. The idea is the following: there exists a simulator algorithm which outputs values that are computationally indistinguishable from the ones that the adversary views during real executions of the protocols of the scheme. This simulator algorithm can be constructed using similar techniques to the ones in the security proofs of the joint generation of discrete-log keys [6] and the threshold Schnorr's signature scheme [20]. Once this simulator is constructed, it is easy to see that a successful attack against our distributed scheme is equivalent to a successful attack against the scheme of Lee *et al.*, and therefore equivalent to a successful attack against the Schnorr's signature scheme. This is computationally infeasible, in the random oracle model.

Thus, if the conditions $\Gamma_A \cap \mathcal{A}_A = \emptyset$ and $\Gamma_B \cap \mathcal{A}_B = \emptyset$ hold, we can state that any subset of $\mathcal{A}_A$ does not obtain any information that allows it to delegate A's signing capability to a proxy entity; basically, this holds because players in B need to receive information from an authorized subset of A in order to compute the shares of the new proxy key. And, by the unforgeability of the distributed Schnorr's signature scheme, any subset of $\mathcal{A}_B$ does not obtain any information that allows it to sign a message on behalf of an original signer entity A.

Steps 3 and 4 in the distributed proxy key generation phase are a variation of Feldman's verifiable secret sharing scheme (which is computationally secure, see [3]). In these steps, players in B detect dishonest players $P_i \in F_{1,A}$ who want to share an incorrect $\tilde{\gamma}_i$ among players in B or who want to give them shares $\tilde{s}_{ij}$ which are inconsistent with the correct γ_i.

Since we impose $\mathcal{A}_A^c \subset \Gamma_A$ and $\mathcal{A}_B^c \subset \Gamma_B$, the scheme is robust: an authorized subset always remains in the set of non rejected players and can execute each step of the protocol.

5 Conclusion and Open Problems

Distributing cryptographic protocols is a way of providing more security and reliability to them. In real systems, the users among which the cryptographic secret task is distributed do not all have necessarily the same power or susceptibility to be corrupted. For this reason, we think that the threshold scenario that is usually considered in distributed cryptography must be extended to a more general framework.

In this work we have adapted to this general framework some threshold protocols related to the verifiable sharing of a random secret value. Namely, we have generalized the verifiable secret sharing scheme of Pedersen [12], the joint generation of discrete-log keys proposed by Gennaro *et al.* in [6] and the threshold Schnorr's signature scheme of Stinson and Strobl [20].

Furthermore, we have used the resulting protocols in the design of a fully distributed proxy signature scheme. In this scheme, a collective entity delegates its signing capability to another collective entity, which will be able to sign documents on behalf of the first one. The secret operations of the scheme (delegation and delegated signature) require the presence of an authorized subset of users of the corresponding entity. The scheme is unforgeable and robust, in the presence of an adversary who corrupts at the same time a tolerated subset of dishonest players in each entity, provided that the structures of the authorized subsets and the tolerated subsets of dishonest players satisfy some combinatorial conditions.

Some problems remain open in the area of distributed proxy signature schemes, specially in the case of non-threshold access structures. Recently the first proxy signature schemes based on RSA have appeared [17], and proxy signatures based on DSS signature scheme could be constructed in a similar way as it has been done with the Schnorr's one. The problem is that distributed versions of RSA [18] and DSS [5] are only known for threshold structures, so designing new distributed proxy signature schemes for general structures would involve solving the open problem of the design of distributed RSA or DSS signature schemes for non-threshold structures.

Finally, the number of secret communications among the participants in our fully distributed scheme is quite large; this fact is in part inherited from the cost of the joint generation of a random secret value. A solution in order to improve this point is to consider *publicly verifiable* protocols as the ones in [15,4], which do not require secret channels among the players, and which can be also extended to the general framework that we have considered in this work. However, these protocols are computationally less efficient, because the number of computations that must perform each player increases. So the use of one model or the other one must be decided according to the requirements, the necessities or the resources of the system.

References

1. G.R. Blakley. Safeguarding cryptographic keys. *Proceedings of the National Computer Conference, AFIPS'79*, pp. 313–317 (1979).
2. E.F. Brickell. Some ideal secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, Vol. **9**, pp. 105–113 (1989).
3. P. Feldman. A practical scheme for non-interactive verifiable secret sharing. *Proceedings of FOCS'87*, IEEE Press, pp. 427–437 (1987).
4. P.A. Fouque and J. Stern. One round threshold discrete-log key generation without private channels. *Proceedings of PKC'01*, LNCS 1992, Springer-Verlag, pp. 190–206 (2001).
5. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. Robust threshold DSS signatures. *Advances in Cryptology-Eurocrypt'96*, LNCS 1070, Springer-Verlag, pp. 354–371 (1996).
6. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Advances in Cryptology-Eurocrypt'99*, LNCS 1592, Springer-Verlag, pp. 295–310 (1999).

7. M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation. *Proceedings of PODC'97*, pp. 25–34 (1997).
8. M. Hwang, I. Lin and E.J. Lu. A secure nonrepudiable threshold proxy signature scheme with known signers. *International Journal of Informatica*, vol. **11**, no. 2, pp. 1–8, (2000).
9. S. Kim, S. Park and D. Won. Proxy signatures, revisited. *Proceedings of ICISC'97*, pp. 223–232 (1997).
10. B. Lee, H. Kim and K. Kim. Strong proxy signature and its applications. *Proceedings of SCIS'01*, Vol. 2/2, pp. 603–608 (2001).
11. M. Mambo, K. Usuda and E. Okamoto. Proxy signatures: Delegation of the power to sign messages. *IEICE Transactions Fundamentals*, Vol. E79-A, No. **9**, pp. 1338–1353 (1996).
12. T.P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. *Advances in Cryptoplogy-Crypto'91*, LNCS 576, Springer-Verlag, pp. 129–140 (1991).
13. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. **13**, Num. **3**, Springer-Verlag, pp. 361–396 (2000).
14. C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, Vol. **4**, pp. 161–174 (1991).
15. B. Schoenmakers. A simple publicly verifiable secret sharing scheme and its applications to electronic voting. *Advances in Cryptology-Crypto'99*, LNCS 1666, Springer-Verlag, pp. 148–164 (1999).
16. A. Shamir. How to share a secret. *Communications of the ACM*, No. **22**, pp. 612–613 (1979).
17. Z. Shao. Proxy signature schemes based on factoring. *Information Processing Letters*, No. **85**, pp. 137–143 (2003).
18. V. Shoup. Practical Threshold Signatures. *Advances in Cryptology-Eurocrypt'00*, LNCS 1807, Springer-Verlag, pp. 207–220 (2000).
19. G. J. Simmons, W. Jackson and K. Martin. The geometry of secret sharing schemes. *Bulletin of the ICA* **1**, pp. 71–88 (1991).
20. D.R. Stinson and R. Strobl. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. *Proceedings ACISP'01*, LNCS 2119, Springer-Verlag, pp. 417–434, (2001).
21. K. Zhang. Threshold proxy signature scheme. *Proceedings of the 1997 Information Security Workshop, Japan*, pp. 191–197 (1997).

Non-interactive Zero-Sharing with Applications to Private Distributed Decision Making

Aggelos Kiayias[1] and Moti Yung[2]

[1] University of Connecticut, CSE, Unit 3155, Storrs, CT USA,
aggelos@cse.uconn.edu
[2] Columbia University, Computer Science, NY, USA
moti@cs.columbia.edu

Abstract. We employ the new primitive of *non-interactive zero-sharing* to realize efficiently and privately various "distributed decision making" procedures. Our methodology emphasizes *non-interactiveness* and *universal verifiability*. Non-interactiveness suggests that there is no bilateral communication between the active participants; instead decision making is achieved by unilateral communication between active participants and a security-wise non-trusted server that participates faithfully in the protocol. Universal verifiability suggests that the participants' actions produce a public audit trail ensuring publicly that they have followed the protocol's specifications. Based on non-interactive zero-sharing, we present constructions for a private veto protocol, a protocol for simultaneous disclosure of information and a privacy-enhancing "plug-in" tool for electronic voting that can be incorporated in homomorphic-encryption based schemes.

Keywords. Distributed Decision Making, Privacy, Veto, Simultaneous Disclosure, Electronic Voting, Proofs of Knowledge.

1 Introduction

"Distributed Decision Making" describes a generic form of a collaborative computation that allows a set of entities to form a common "decision" that takes into account the inputs of each individual entity. A distributed decision making protocol should satisfy a number of requirements (security, privacy, efficiency, trustworthiness) and be executable under various constraints (communication topology, connectivity, synchronicity, adversarial model).

A property that is crucial in many settings and is the most interesting from a cryptographic viewpoint is that of *privacy* of the individual inputs of the participating entities. That is, decision making should be achieved without requiring individual entities to reveal their contributions to the procedure.

Some instances of distributed decision making procedures can be viewed as a special case of *secure multiparty computation*, where players wish to contribute private inputs to a publicly known function where the function is publicly evaluated without revealing the private inputs. Despite the fact that it is possible

R.N. Wright (Ed.): FC 2003, LNCS 2742, pp. 303–320, 2003.
© Springer-Verlag Berlin Heidelberg 2003

to realize any secure multiparty computation protocol using generic techniques [GMW87], such protocols are mere plausibility results since they are not practical. As a result, the trend is to isolate specific secure multiparty computation instances that are important in practice (e.g., e-voting), and concentrate on the implementation of protocols that realize such instances in a direct and efficient fashion.

Apart from the privacy of the individual inputs, various other properties can be crucial to the practical realization of a distributed decision making protocol. As an example consider the property of minimized interaction among the active participants. Indeed, bilateral communication between individual entities that participate in a multiparty scheme has the potential of introducing several computational and communication problems (e.g., the issue of pairwise disputes regarding a communication transcript along a link).

Server aided secure multiparty computation, introduced in [Bea97], showed that one can adopt communication patterns that agree with the way network communication is abstracted (the "client-server" approach) and achieve secure multiparty computation assuming a majority of honest participating servers. Nevertheless, in order to ensure the practicality of multiparty protocols, still, one has to isolate a specific multiparty computational task and realize it in a direct way so that various efficiency and security requirements can be met.

The concentration on the practical realization of specific secure multiparty protocols and specifically of electronic voting [Cha81,CF85,Cha88,FOO92,BT94] [SK94,CGS97,Sch99,HS00,KY02], introduced various useful protocol concepts that have many applications beyond the e-voting domain. Motivated by this, in this work we concentrate on a new basic primitive called *non-interactive verifiable zero-sharing* that can be a basic building block in realizing various distributed decision making procedures that extend beyond the e-voting scenario; this primitive was implicitly used first in [KY02] in the context of boardroom elections, but our goal here is to isolate it as a primitive. Our constructions emphasize non-interactiveness and universal verifiability as fundamental properties for the practical implementation of distributed decision making protocols:

1. Non-interactiveness suggests that there is no bilateral communication between the active entities in the scheme; the computation is performed through the unidirectional communication of the entities with a passive non-trusted server that acts as a bulletin board (a concept that was introduced in the context of e-voting schemes in [CF85]).
2. Universal verifiability, initially introduced for the tallying phase of e-voting schemes (see e.g. [CFSY96]), suggests that the active participants' actions generate a public audit trail that allows third parties to verify that the participating entities follow the specifications of the protocol. The basic tools for such verifiability are non-interactive proofs of knowledge, see e.g. [CDS94, DDPY94].

Armed with the primitive of non-interactive zero-sharing, we show how some basic distributed decision making procedures can be realized efficiently, satisfying

non-interactiveness, universal verifiability and privacy of individual inputs. We present three explicit protocol constructions for the following problems:

- **Private Veto**. A private veto protocol is a basic *consensus* mechanism that allows a set of entities to test whether there is general agreement on a given decision. Observe that from a privacy viewpoint this is a different task compared to "yes/no" voting as the result should merely reveal whether there is consensus, rather than an individual count of the number of agreeing and disagreeing entities. We present a concrete protocol construction for the specification we formalize and prove it to be secure.
- **Simultaneous Disclosure**. In various settings where a set of agents is required to submit some suggestion on a given matter, it is of crucial importance to facilitate the simultaneous disclosure of all proposals. Observe that privacy per se is not a concern here, rather what we are interested to achieve is that all proposals remain private until every contributing entity submits its proposal, and after this fact all proposals should be revealed *without* the participation of any of the contributing parties. As a side note, we remark that using a (self-controlled) commitment scheme is not a satisfactory solution to this problem, since an entity might refuse to submit its decommitment. Instead, our construction allows the "decommitment" phase to be performed without the "help" of the participating entities.
- **E-voting with enhanced privacy.** In the e-voting domain, achieving strong voter privacy is a very challenging goal, especially in the large-scale setting. In the current state of the art e-voting schemes, voter privacy relies on honesty of a quorum of authorities, namely assumptions of the form "a certain number of authorities do not collude against the privacy of the voters." Although sufficient in some settings, these assumptions do not capture voter privacy as it is understood ideally. Here we present a "voting utility" that can be seamlessly integrated in large scale "homomorphic encryption" based e-voting schemes, and enables a set of voters to protect the privacy of their votes, even in settings where all authorities may be dishonest and try to violate voters' privacy.

The organization of the paper: in section 2 we define our model, called *private distributed non-interactive decision-making*. In section 3 we present the basic tools and the zero-sharing protocol. Finally, in sections 4, 5 and 6 we present our implementations of the Private Veto protocol, the Simultaneous Disclosure protocol, and the E-voting Privacy Enhancing Utility respectively.

2 Private Non-interactive Distributed Decision Making

A private non-interactive decision-making protocol involves a number of entities $B_1, \ldots, B_n$ and a (non-trusted, in the sense of privacy) server A that assists the entities in the course of the protocol. We assume that the entities can communicate with the server A in an authenticated form. On the other hand, our

protocols do not assume private channels between the server and the participants. A plays the role of a bulletin board server (see section 3.1) which allows parties to publish information from authenticated sources.

The execution of a private non-interactive decision-making protocol is broken into a number of *rounds* where in each round each participant B_i communicates a value to the server A — note that no bilateral communication is permitted. Our communication model is *semi-synchronous*: the order and concurrency of the participants' actions is not important within the course of a round, however the participants need to be active within a certain round time-out period. Note that at any round and, in fact at any time, the content of the bulletin board A is readable by anyone.

After the end of each round, the server A might perform a publicly-verifiable ciphertext processing over the submitted ciphertext values. This is a public computation that can be verified simply by repeating it. Its purpose is to alleviate the computational costs of the participants.

Finally, each private non-interactive decision making protocol has a *decision producing procedure* $\mathcal{T}$ that given the public-transcript of the protocol it outputs the decision. Purposedly we do not detail the properties of $\mathcal{T}$ as they might vary depending on the specific application.

A variety of properties needs to be satisfied by a private non-interactive distributed decision making protocol. We describe them informally below:

- **Security/Privacy.** It should be infeasible for an adversary that is controlling a number of entities to reveal the contribution of certain entities.
- **Fairness.** It should be infeasible for an entity to compute the distributed decision prior to publishing its individual decision.
- **Universal Verifiability.** It should be possible for any third party to verify that the entities are following the protocol as specified.
- **Batching Property**. Each entity after making its decision will only engage in a single round of communication with the server and then the protocol will terminate (i.e. the "contribution dependent" rounds are limited to a single –final– round). This suggests that the protocol is comprised by a number of "preparatory" rounds that can be executed ahead of time and stored by the server for later usage (batching). Private non-interactive decision making protocols that satisfy the batching property have optimal "on-line" round complexity (where online refers to the number of rounds required from the time an entity is required to make its decision till the time that the distributed decision is announced).

3 Basic Tools

The participants in the protocol are n entities denoted by $B_1, \ldots, B_n$ and the bulletin board server. Each entity has a unique identification string denoted by $I(B_j)$. Identification strings are publicly known, and can be in the form of pseudonyms (depending on the level of privacy required).

3.1 The Bulletin Board

A bulletin board is a basic primitive which we employ for all the necessary communication between the parties that participate in our protocols. The bulletin board was introduced in the context of e-voting in [CF85]. It is a public-broadcast channel with memory. Any party (even third-parties) can read information from the bulletin board. Writing on the bulletin board by the active parties is done in the form of appending data in a specially designed area for each party. Erasing from the bulletin board is not possible, and appending is verified so that any third party can be sure of the communication transcript. In the sequel, the phrase "party X publishes value Y" means that X appends Y to the portion of the bulletin board that belongs to X. The bulletin board authority (server) might participate in the protocol to alleviate the computational costs of the participants and administer the protocol in general. Server-based ciphertext processing helps in reducing the computations of the parties, whenever trusted. Computation performed by the server will be publicly verifiable (e.g., by repeating the computation whenever not trusted). The bulletin board server is also responsible for administering the protocol, namely, it performs actions such as starting and terminating the decision-making procedure, and maintaining a registry of the eligible entities that should gain access to the bulletin board.

3.2 Intractability Assumption

Let $\mathcal{G}_k$ be some family of groups, where the order of each $G \in \mathcal{G}_k$ is exponential in k and so that solving the Decisional Diffie-Hellman problem in groups of $\mathcal{G}_k$ is hard:

Definition 1. DDH. *Fix some $G \in \mathcal{G}_k$. Consider quadruples of the form* R, *$\langle g, g^a, g^b, g^c \rangle$ with $a, b, c < \text{order}(g)$ and quadruples of the form* D, *$\langle g, g^a, g^b, g^{ab} \rangle$ with $a, b < \text{order}(g)$, where g is a generator of G. A predicate solves the DDH problem if it can distinguish the collection* D *from the collection* R.

The DDH-Assumption *for $\mathcal{G}$ suggests that any predicate that solves the DDH problem succeeds with probability that differs from $\frac{1}{2}$ by at most a negligible in k fraction.*

For example, a family of groups $\mathcal{G}_k$ over which DDH is assumed to be hard is the following: the family of all groups G such that p, q are large primes with $q \mid p - 1$ and G is the unique subgroup of $\mathbf{Z}_p^*$ of size q; here the parameter k is the number of bits of p, q. Now fix some family $\mathcal{G}_k$. Let *Gen* be a probabilistic polynomial-time algorithm that given 1^k generates the description of a group $G \in \mathcal{G}_k$, and two random elements from G, g, h (with relative discrete logs unknown); this can be also be produced distributively, see [GJKR99]. We will denote the order of G by q. Observe that arithmetic in the exponents of elements in G is performed in $\mathbf{Z}_q$.

We assume that all parties, either observe *Gen* with fixed coin-tosses on some public true random string, or are using a suitable cryptographic protocol for generating shared randomness and subsequently execute *Gen*. In both cases the description of the group and the elements g, h will be available to all parties.

3.3 Proofs of Knowledge

Proofs of knowledge is a fundamental tool for ensuring that entities follow the specifications of a Cryptographic protocol, and in the non-interactive case they produce an auditable public trail of the participants' actions. Here we take advantage of a non-interactive proof of knowledge of equality of discrete-logs over a different base in the same group. Using a proof of knowledge introduced in [CP93], this is possible as described in figure 1. Note that this protocol is proven to be zero-knowledge only in the case of a honest verifier (see e.g. [CGS97]), but this is sufficient in our setting.

Prover		Verifier
publishes $\langle R, R' \rangle$		
$w \in_R \mathbf{Z}_q$		
$a := g^w, b := h^w$	$\xrightarrow{a,b}$	
	$\xleftarrow{c}$	$c \in_R \mathbf{Z}_q$
$r := w + sc (\mathrm{mod} q)$	$\xrightarrow{r}$	$g^r \stackrel{?}{=} a(R)^c$
		$h^r \stackrel{?}{=} b(R')^c$

Fig. 1. Let g, h be two bases of order q, and $R = g^s, R' = h^s$. The protocol above shows how to prove that $\log_g R = \log_h R'$

The well-known Fiat-Shamir heuristics [FS87] can be used to make the proof non-interactive and ensure that the challenge c is chosen "honestly": if $\mathcal{H}$ is a cryptographically strong hash function (thought of as a random oracle), then c is defined as $\mathcal{H}(I(B), R, R', a, b)$, where $I(B)$ is a publicly known string that identifies the prover. Subsequently the prover publishes $\langle R, R', c, r \rangle$; the verifier performs the following test: $c \stackrel{?}{=} \mathcal{H}(I(B), R, R', g^r R^{-c}, h^r (R')^{-c})$, which can be easily shown to be equivalent to the two tests given in figure 1. This method ensures that the challenge c is chosen at random (under the assumption that $\mathcal{H}$ is a random oracle hash). When c is chosen using a random oracle hash we will denote the sequence $\langle R, R', c, r \rangle$ defined as above for the bases g, h by $\mathsf{PKEQDL}[x : (R = g^x) \wedge (R' = h^x)]$.

3.4 Non-interactive Verifiable Zero-Sharing

In this section we describe the Non-Interactive Verifiable Zero-Sharing protocol. This protocol appeared in [KY02] as a phase of a *small-scale* (boardroom) e-voting design. In the present work we treat this protocol as a general tool for building distributed decision making protocols. The protocol is divided in three rounds of interaction with the server.

In the first round, every entity B_i selects a random value $\alpha_i \in \mathbf{Z}_q$ and publishes $h_i := h^{\alpha_i}$ (the entity's personal generator for G).

In the second round, each entity B_i, selects n random values $s_{i,j} \in \mathbf{Z}_q$, $j = 1, \ldots, n$, such that $\sum_{j=1}^{n} s_{i,j} = 0$. Each B_i then, publishes the pairs $\langle R_{i,j}, R'_{i,j} \rangle$,

s.t. $R_{i,j} := g^{s_{i,j}}$ and $R'_{i,j} := h_j^{s_{i,j}}$. Entity B_i should prove to any third party that $\log_g R_{i,j} = \log_{h_j} R'_{i,j}$, and this can be achieved by publishing the non-interactive proof of knowledge $\mathsf{PKEQDL}[x : (R_{i,j} = g^x) \wedge (R'_{i,j} = h_j^x)]$.

The server for each $j = 1, \ldots, n$ computes the products $R_j := \prod_{i=1}^n R'_{i,j}$ and $R'_j := \prod_{i=1}^n R'_{i,j}$ and publishes them on the board (see figure 2). Subsequently, in the final round, each entity B_j reads the value R'_j from the server.

Observe that if $t_j := \sum_{i=1}^n s_{i,j}$ it holds that (i) $\sum_{j=1}^n t_j = 0$, and (ii) $(R'_j)^{\alpha_j^{-1}} = h^{t_j}$ for all $j = 1, \ldots, n$. The quantity h^{t_j} will be the *exponentiated zero-share* of the entity B_j and (under the DDH assumption) it is only available to the entity B_j.

$R_{i,j}$:

B_1	:	$g^{s_{1,1}}$	$\ldots$	$g^{s_{1,n}}$
B_2	:	$g^{s_{2,1}}$	$\ldots$	$g^{s_{2,n}}$
$\vdots$		$\vdots$	$\ldots$	$\vdots$
B_n	:	$g^{s_{n,1}}$	$\ldots$	$g^{s_{n,n}}$
Server		$R_1 = g^{t_1}$	$\ldots$	$R_n = g^{t_n}$

$R'_{i,j}$:

B_1	:	$h_1^{s_{1,1}}$	$\ldots$	$h_n^{s_{1,n}}$
B_2	:	$h_1^{s_{2,1}}$	$\ldots$	$h_n^{s_{2,n}}$
$\vdots$		$\vdots$	$\ldots$	$\vdots$
B_n	:	$h_1^{s_{n,1}}$	$\ldots$	$h_n^{s_{n,n}}$
Server		$R'_1 = h_1^{t_1}$	$\ldots$	$R'_n = h_n^{t_n}$

Fig. 2. The contents of the Bulletin Board after the second round.

When the zero-sharing stage is completed the bulletin board server signs the bulletin board and stores its contents for later usage. The following result, justified in [KY02], describes the properties of the non-interactive zero-sharing protocol:

Fact 1 *At any time, after the completion of the non-interactive zero-sharing,*
(i) Any third-party can verify that $\log_g R_{i,j} = \log_{h_j} R'_{i,j}$*, for any* i, j*.*
(ii) Any third-party can verify that $\sum_{j=1}^n s_{i,j} = 0$*, for any* i*.*
(iii) If at least one entity chose the $s_{i,j}$ *values at random, then the values* $t_j := \sum_{i=1}^n s_{i,j}$ *are random elements of* $\mathbf{Z}_q$ *with the property* $\sum_{j=1}^n t_j = 0$*.*

In our protocols, we will employ non-interactive verifiable zero-sharing as the preparatory rounds before the actual decision making takes place.

4 A Private Veto Protocol

4.1 Definitions: Correctness and Security

A private veto protocol involves a number of entities $B_1, \ldots, B_n$ that decide whether or not to veto a publicly known proposal. Every participant B_i publishes a value D_i, and there is a publicly available "veto testing" predicate $\mathcal{T}$ with the property

$$\mathcal{T}(D_1, \ldots, D_n) = 1 \text{ iff } \exists i \;\; B_i \text{ vetoes}$$

In fact, we will use a more generalized correctness definition where the veto testing predicate is allowed to fail with very small (negligible) probability, i.e.

$\mathbf{Prob}[(\mathcal{T}(D_1,\ldots,D_n)=0)\wedge(\exists i\ \ B_i\ \text{ vetoes})]$ should be negligible in the security parameter.

The probability is taken over all the internal coin tosses of the participants that were used to generate the values $D_1,\ldots,D_n$. Because of the distributed nature of the computation, the correctness definition has to consider the fact that some participants might be misbehaving:

Definition 2. (Veto Protocol Correctness) *Provided that at least one entity is honestly following the protocol (independently of its decision), the probability*

$$\mathbf{Prob}[(\mathcal{T}(D_1,\ldots,D_n)=0)\wedge(\exists i\ \ B_i\ \text{ vetoes})]$$

is negligible, where the probability is taken over all the internal coin tosses of the participating entities.

The security property that we want to achieve has to do with the privacy of the participants. In particular, given that a certain proposal is vetoed, it should be impossible for an adversary to distinguish which users vetoed.

We formalize this form of security as follows:

Definition 3. (Security: Privacy of Decisions) *Let $\mathcal{A}$ be an adversary that controls a number of entities $k \leq n-2$. There are at least two entities, say B_1, B_2, the adversary does not control. The participants execute the veto protocol, and entity B_1 vetoes the proposal. Let $\mathtt{t}^{\mathrm{V}}_{B_2}$ be the random variable that describes protocol transcripts generated as above with B_2 vetoing the proposal, and $\mathtt{t}^{NV}_{B_2}$ be the random variable that describes protocol transcripts with B_2 agreeing with the proposal. If the distinguishing probability of the adversary between $\mathtt{t}^{\mathrm{V}}_{B_2}$ and $\mathtt{t}^{NV}_{B_2}$ is a negligible function, we say that the veto protocol is secure.*

We remark that the above definition of security (which suggests that given that an entity vetoes it is not possible to distinguish whether a second entity vetoes or not) can be easily seen to imply the security property that mandates the following: given that there is a veto decision by one of two entities it is impossible to distinguish which one vetoed (the reduction can be accomplished using a standard triangular inequality argument).

Regarding the relationship of vetoing with e-voting, observe that implementing a veto protocol with a secure yes/no voting protocol fails the security definition (this is because the yes/no voting procedure will reveal the number of agreeing and disagreeing entities).

We will also consider the properties of fairness and batching as they were described in section 2.

4.2 The Veto Protocol

The entities $B_1,\ldots,B_n$ perform the non-interactive zero-sharing protocol as described in section 3.4. After the end of the protocol the bulletin board contains the public personal generators $h_1,\ldots,h_n$ of each participant as well as the values $R'_1,\ldots,R'_n$. Recall that $\sum_{i=1}^{n}\log_{h_i} R'_i = 0$.

The server publishes the motion to be unanimously decided by the participants and then signals the beginning of the procedure. Each entity B_i publishes a value D_i that, depending on whether B_i wishes to veto the proposal, is defined as follows:

$$D_i = \begin{cases} (R'_i)^{\log_h(h_i)^{-1}} & \text{agreement} \\ \text{random} \in G & \text{veto} \end{cases}$$

The veto testing predicate $\mathcal{T}$ is defined as follows:

$$\mathcal{T}(D_1, \ldots, D_n) = \begin{cases} 1 \text{ if } \prod_{i=1}^{n} D_i = 1 \\ 0 \text{ otherwise} \end{cases}$$

Observe that the correctness of the veto protocol, as stated in definition 2, follows easily. The proposition below completes the description of the veto protocol. The fairness and security of the protocol are treated separately in sections 4.3, and 4.4 respectively.

Proposition 1. *The veto protocol is a 3-round private non-interactive decision making protocol that satisfies the batching property.*

The batching property suggests that one can store many instantiations of the zero-sharing protocol, and then, whenever there is a certain motion that needs to be decided by the participants, this can be done in a single round of unilateral communication.

4.3 Fairness: Administrating and Terminating the Protocol

The bulletin board server is responsible for the administration of the veto protocol. It is imperative that the server prevents reading of the decisions as the protocol progresses in order to ensure fairness. This is because the last entity to publish its D-value is able to compute $\mathcal{T}$ before making its decision and publishing it. Fairness can be ensured as follows: the server participates in the zero-sharing stage acting as one of the entities, say entity i_0. Naturally we should require from the server to agree with the proposal in a publicly verifiable manner. As a result it will publish $D_{i_0} := (R'_{i_0})^{\alpha_{i_0}^{-1}}$ together with the non-interactive proof of knowledge PKEQDL$[x : (h = h_{i_0}^x) \wedge (D_{i_0} = (R'_{i_0})^x)]$ that ensures that the correct value is published. The server also signs the contents of the bulletin board, thus officially terminating the protocol. Given the way the entities' decisions are formed and fact 1 it easy to see that:

Proposition 2. *The administration of the election by the Bulletin Board Authority as described above ensures fairness.*

We remark that fairness relies on the honesty of the server. This is unavoidable in the communication model we consider (cf. the case of secret-sharing schemes where a similar "magic-box" mechanism needs to be employed to prohibit the last user to publish a share from computing the secret prior to submitting its share). Nevertheless the fairness dependency on the server can be relaxed as our scheme allows the distribution of the server using standard threshold techniques, e.g. [GJKR99], (and then fairness will rely in a "honest majority" argument).

4.4 Security

We start with a lemma that will be useful in the reduction in the proof of security.

Lemma 1. *Consider tuples of the form* $\mathtt{D}'$, $\langle g, h, h_1, h_2, h_1^{-b}, h_2^b, g^b, h^b\rangle$ *and tuples of the form* $\mathtt{R}'$, $\langle g, h, h_1, h_2, h_1^{-b}, h_2^b, g^b, R\rangle$ *where* R *is chosen at random from* G. *Given that DDH is hard over* G, *it holds that* $\mathtt{D}'$ *and* $\mathtt{R}'$ *are indistinguishable for any probabilistic polynomial-time bounded observer.*

Proof. Let $\langle g, A, B, C\rangle$ be a challenge for the DDH. Consider the following tuple

$$\mathsf{C} := \langle g, A, g^{\lambda_1}, g^{\lambda_2}, B^{-\lambda_1}, B^{\lambda_2}, B, C\rangle$$

where $\lambda_1, \lambda_2 < \mathrm{order}(g)$. It is easy to verify that if $\langle g, A, B, C\rangle \in \mathtt{D}$ it holds that $\mathsf{C} \in \mathtt{D}'$ and that if $\langle g, A, B, C\rangle \in \mathtt{R}$ it holds that $\mathsf{C} \in \mathtt{R}'$. The result of the lemma follows immediately. □

Theorem 1. *Our veto protocol described in section 4.2 is secure according to definition 3.*

Proof. Suppose that $\mathcal{A}$ is an adversary that breaks the security of the veto protocol as defined in the security definition 3. Without loss of generality assume that the adversary controls all entities $B_3, \ldots, B_n$.

First observe that due to the proofs of knowledge required in the non-interactive zero-sharing phase for any $i = 3, \ldots, n$, the property $\sum_{j=1}^{n} \log_{h_j} R'_{i,j} = 0$, is enforced and as a result $\mathcal{A}$ cannot disrupt the zero-sharing property of the protocol without being detected.

Each of the players $B_3, \ldots, B_n$ publish their decisions $D_3, \ldots, D_n$, and subsequently B_1 publishes D_1 chosen at random from G, and B_2 publishes its decision D_2. The adversary $\mathcal{A}$ distinguishes protocol transcripts in which B_2 vetoes from those that B_2 does not veto.

We will show how to simulate the protocol using the adversary $\mathcal{A}$ to distinguish the two distributions $\mathtt{D}'$ and $\mathtt{R}'$, defined in lemma 1.

Let $\mathsf{C} := \langle g, h, h_1, h_2, H'_1, H_2, G_*, D_*\rangle$ be a challenge for a distinguisher of the two distributions $\mathtt{D}'$ and $\mathtt{R}'$.

We control entities B_1, B_2 and the random-oracle that is given to the adversary. First we have B_1, B_2 publish h_1, h_2 as the public generators of G. In the zero-sharing phase we have the entity B_2 publish the values

$$\begin{array}{lll}
R'_{1,1} = h_1^{s_{1,1}} H'_1 & R_{1,1} = g^{s_{1,1}} (G_*)^{-1} & \mathsf{PK}_1 \\
R'_{1,2} = h_2^{s_{1,2}} H_2 & R_{1,2} = g^{s_{1,2}} (G_*) & \mathsf{PK}_2 \\
R'_{1,3} = h_3^{s_{1,3}} & R_{1,3} = g^{s_{1,3}} & \mathsf{PK}_3 \\
\vdots & \vdots & \vdots \\
R'_{1,n} = h_n^{s_{1,n}} & R_{1,n} = g^{s_{1,n}} & \mathsf{PK}_n
\end{array}$$

where $s_{1,1} + \ldots + s_{1,n} = 0 (\mathrm{mod} q)$ and are selected (otherwise) at random from $\mathbf{Z}_q$. Observe that the non-interactive proofs of knowledge PK_1 and PK_2 refer to values of which the simulator is ignorant of their discrete-logarithm and as result

they have to be simulated; this is achieved as follows: given two values v_1, v_2 for which we wish the simulator to generate a proof of knowledge of equality of the discrete-log w.r.t. the bases h, g we select random c, r and then we write the proof of knowledge $\langle v_1, v_2, c, r\rangle$ additionally we set $\mathcal{H}(I(B_2), v_1, v_2, v_1^{-c}g^r, v_2^{-c}h^r) = c$ (i.e. we record this in the table of values for the random oracle $\mathcal{H}$). The simulation of PK_1 and PK_2 is executed prior to starting the adversary so that we have already selected the necessary entries in the random-oracle table (subsequent random-oracle queries of the adversary are simulated at random).

The entity B_1 follows the non-interactive zero-sharing phase as specified in the protocol's description. Finally, we have the entity B_1 publish a random value as its decision and entity B_2 publish D_* as its decision.

It is easy to verify that if C was selected from the distribution $\mathtt{D}'$ the simulator will produce a valid protocol transcript in which the entity B_2 agrees with the proposal, whereas when C is selected from the distribution $\mathtt{R}'$ the simulator will produce a valid protocol transcript in which the entity B_2 vetoes the proposal. If the adversary is capable of distinguishing the two protocol transcripts it is immediate that we can also distinguish the two distributions $\mathtt{R}'$ and $\mathtt{D}'$ something that violates the DDH assumption according to lemma 1. □

Remark. It is clear from the above that any distinguisher that breaks the security of our veto protocol with probability of success α will also break DDH with success probability α. If we allow the adversary to select the two users B_1, B_2 that will be controlled by the simulator *after* the execution of the zero-sharing phase (so we allow a little more power to the adversary) then the simulator will break the DDH with probability $\alpha \cdot \frac{1}{t^2-t}$ where t is the number of entities that are not controlled by the adversary (note that if α is non-negligible then $\alpha \cdot \frac{1}{t^2-t}$ will also be non-negligible).

5 Simultaneous Disclosure

In a standard scenario in distributed decision making, each member of a group wishes to contribute something for discussion, such as an offer, or a request. However none of the members is willing to put his contribution in the table first, as this will give the advantage to some members of the group to withhold or modify their contribution if they wish prior to submitting it. In "Simultaneous Disclosure", we want to design a protocol that allows to a set of parties to submit their contributions in such a way so that all will be disclosed at the same time.

Observe that a solution based on a commitment scheme is not satisfactory in this setting as it allows entities to refuse to decommit depending on the decommitted proposals up to their turn.

5.1 Definitions: Correctness and Security

A simultaneous disclosure protocol involves a number of entities $B_1, \ldots, B_n$ with each B_i submitting a contribution c_i. Every participant B_i will publish a string D_i, and there is a publicly known extraction algorithm $\mathcal{T}$ with the property

$$\mathcal{T}(D_1, \ldots, D_n) = c_1 || c_2 || \ldots || c_n$$

The security property that we want to achieve suggests that using any proper subset of $\{D_1, \ldots, D_n\}$, and even in the case that we control the actions of some of the entities that participate, the contributions of the remaining entities prior to the termination of the protocol are indistinguishable from random. We formalize this form of security as follows:

Definition 4. (Security: Simultaneous Disclosure) *Let $\mathcal{A}$ be an adversary that controls a number of entities $k \leq n-2$. The adversary does not control the last entity, say B_n, to submit the last encrypted contribution D_n, and some other entity B_{i_0}. Let $\mathsf{t}_{B_{i_0}}^{c_{i_0}}$ be the random variable that describes portions of protocol transcripts prior to B_n's final move, with B_{i_0} entering the public contribution c_{i_0}, and $\mathsf{t}_{B_{i_0}}^{\mathsf{random}}$ be the random variable that describes portions of protocol transcripts prior to B_n's final move, where B_{i_0} enters a random contribution. If the distinguishing probability of the adversary between $\mathsf{t}_{B_{i_0}}^{c_{i_0}}$ and $\mathsf{t}_{B_{i_0}}^{\mathsf{random}}$ is a negligible function, we say that the Simultaneous Disclosure protocol is secure.*

Note that the restriction that the adversary is not controlling the last entity to submit a contribution is unavoidable, since after the last contribution the specification of simultaneous disclosure mandates that all contributions are revealed (i.e. the privacy requirement is "lifted").

The above security definition can be seen as a game where the adversary needs to distinguish whether the contribution of a certain entity is random or not, and can easily be extended to related settings, e.g. in cases where the adversary is required to distinguish whether a set of entities are publishing random contributions or not (this can be shown by a standard "hybrid" argument based on the theorem above), or in cases where the adversary is required to distinguish which one of two possible contributions a certain entity is submitting (which can be shown using the theorem above and the triangular inequality).

We will also consider the properties of fairness and batching as they were described in section 2.

5.2 The Simultaneous Disclosure Protocol

The entities $B_1, \ldots, B_n$ perform n independent instantiations of the non-interactive zero-sharing protocol as described in section 3.4. We remark that these instantiations can be executed in a concurrent manner: there is no need to add additional rounds of communication between participants and the server.

After the end of the protocol the bulletin board contains the public personal generators $h_1[\ell], \ldots, h_n[\ell]$ of each participant as well as the values $R'_1[\ell], \ldots, R'_n[\ell]$ for the ℓ-th instantiation where $\ell \in \{1, \ldots, n\}$.

We will assume that the contribution of each entity B_i belongs to the group G (i.e. there is an embedding of the set of all possible contributions into G). If contributions are lengthy, participants might simply encrypt their actual contributions with a block-cipher such as AES, and then enter as their contribution in the simultaneous disclosure protocol the block-cipher encryption key.

We remark that it is to up to the participants to contribute "meaningful" contributions. We don't impose any notion of "well-formed" contributions, since we opt for a generic design; instead what we want to make sure is that the decision of a participant to submit a meaningful contribution or not should *not* depend on the actions of other participants.

After the initial phase of the zero-sharing, the i-th entity B_i publishes the value $D_i := \langle D_{i,1}, \ldots, D_{i,n} \rangle$ defined as

$$\Big\langle (R'_i[1])^{(\alpha_i[1])^{-1}}, \ldots, (R'_i[i-1])^{(\alpha_i[i-1])^{-1}}, c_i \cdot (R'_i[i])^{(\alpha_i[i])^{-1}},$$

$$(R'_i[i+1])^{(\alpha_i[i+1])^{-1}}, \ldots, (R'_i[n])^{(\alpha_i[n])^{-1}} \Big\rangle$$

Additionally each entity B_i publishes the non-interactive proofs of knowledge $\text{PKEQDL}[x : (h[j] = (h_i[j])^x) \wedge (D_{i,j} = (R'_i[j])^x)]$ for all $j \in \{1, \ldots, n\} - \{i\}$.

The disclosure operation is achieved with the following extraction algorithm:

$$\mathcal{T}(D_1, \ldots, D_n) := \langle \prod_{i=1}^{n} D_{i,1}, \ldots, \prod_{i=1}^{n} D_{i,n} \rangle = \langle c_1, \ldots, c_n \rangle$$

The proposition below completes the description of the simultaneous disclosure protocol. The fairness and security of the protocol are treated separately in sections 5.3, and 5.4 respectively. It is easy to see that the simultaneous disclosure protocol satisfies the batching property: many executions of n zero-sharing rounds can be performed and stored for later usage. When the appropriate time comes, the participants simply publish the tuple D_i in order to perform simultaneous disclosure.

Proposition 3. *The simultaneous disclosure protocol is a 3-round private non-interactive decision making protocol that satisfies the batching property.*

5.3 Fairness: Administrating and Terminating the Protocol

Similarly as in section 4.3, fairness in the simultaneous disclosure protocol, can be ensured as follows: the server participates in all zero-sharing stages acting as one of the entities, say entity i_0. Naturally, the server will submit a predetermined contribution which we will assume that it is equal to 1. As a result it will publish $D_{i_0} := \langle D_{i_0,1}, \ldots, D_{i_0,n} \rangle$ defined as

$$\Big\langle (R'_{i_0}[1])^{(\alpha_{i_0}[1])^{-1}}, \ldots, (R'_{i_0}[n])^{(\alpha_{i_0}[n])^{-1}} \Big\rangle$$

together with the non-interactive proofs of knowledge $\text{PKEQDL}[x : (h[j] = (h_{i_0}[j])^x) \wedge (D_{i_0,j} = (R'_{i_0}[j])^x)]$ for all $j = 1, \ldots, n$. This ensures that the server follows the protocol and publishes the contribution 1. The server also signs the contents of the bulletin board, thus officially terminating the protocol.

Given the way the entities' decisions are formed and fact 1 it easy to see that:

Proposition 4. *The administration of the election by the Bulletin Board Authority as described above ensures fairness.*

As in section 4.3, fairness relies on the behavior of the server. Again, this dependency can be relaxed by employing standard threshold techniques that will distribute the capability of the server in a number of authorities. The straightforward integration of such threshold techniques in our design is a main advantage compared to the trivial solution to the simultaneous disclosure problem, where each participant commits to its contribution and then communicates the decommitment information privately to the server.

5.4 Security

The security of our scheme is based on a similar simulation of the zero-sharing scheme that was utilized in the proof of theorem 1. Specifically, we show:

Theorem 2. *Our simultaneous disclosure protocol described in section 5 is secure according to definition 4.*

Proof. (sketch) Suppose that $\mathcal{A}$ is an adversary that breaks the security of the simultaneous disclosure protocol. Without loss of generality assume that the adversary controls all entities $B_1, \ldots, \ldots, B_{i_0-1}, B_{i_0+1}, \ldots, B_{n-1}$, and entity B_n refrains from publishing the D_n tuple. The adversary $\mathcal{A}$ distinguishes with non-negligible probability protocol transcripts in which the entity B_{i_0} uses a fixed public contribution c_{i_0} from protocol transcripts in which the entity B_{i_0}, uses a random contribution.

Let $\mathsf{C} := \langle g, h, h_{i_0}, h_n, H'_1, H_2, G_*, D_* \rangle$ be a challenge for a distinguisher of the two distributions $\mathtt{D}'$ and $\mathtt{R}'$.

The simulator publishes the values h_{i_0}, h_n as the public generators for the entities B_{i_0} and B_n, for the i_0-th execution of the zero-sharing protocol. Then it simulates the zero-sharing protocol following the same techniques used in the proof of theorem 1. When the turn of the entity B_{i_0} comes (which is controlled by the simulator) it publishes the tuple D where in the i_0-th location the value D_* is placed. If the adversary is capable of breaking the security of the simultaneous disclosure scheme with non-negligible success then it follows from lemma 1 that the DDH Assumption is violated. □

6 Ballot Secrecy in Large Scale E-Voting

Electronic voting is perhaps the most widely studied example of a private distributed decision making procedure. The privacy of the participants in an election is naturally a fundamental property. In present real world systems, ballot secrecy is ensured through physical means. The "simulation" of this physical concealment of the ballot in the e-voting domain proved to be a very challenging task. This is due to the fact that ballots, if are to be concealed somehow, should nevertheless be "sufficiently" accessible for the purpose of tallying, checking integrity, and revealing the final results.

The standard solution that has been established in the literature (see e.g. [BY86]) in order to deal with the issue of ballot secrecy is to introduce a set of authorities, that distributedly share the capability of accessing the contents of the ballots (or linking a certain voter to his cast ballot, if the scheme is based on a cryptographic mix-net). Typically there is a threshold, that the number of conceding authorities needs to exceed in order to enable the capability to access the ballots' contents. In the case that some authorities are "corrupted" (in the sense that they wish to violate the privacy of a certain set of voters), they are capable of doing so provided that their number exceeds the threshold. Raising the threshold very high is very expensive since it induces dramatic effects in the efficiency and robustness of the protocol, two properties that can be of crucial importance, especially in the large-scale setting. While this type of voter privacy might be satisfactory in some settings, it has its shortcomings and raises serious privacy concerns (e.g. as expressed by Brands, [Bra99]).

By employing the basic tool of non-interactive zero-sharing, we propose a solution for enhanced privacy for voting systems that does not depend on the underlying threshold structure employed in the election scheme. Our approach is based on the development of a *voter-controlled privacy mechanism*, that allows to groups of interested voters that participate in a large scale e-voting scheme to enhance their privacy at the cost of performing our zero-sharing protocol. As a result, our solution can be seen as a "voting utility" that can be initiated by small subsets of the voter population that are interested in enhanced ballot secrecy. The objective of the utility is to provide (computational) ballot secrecy as it is understood in the ideal physical elections setting. Namely, consider a precinct where all voters apply the utility, then a person's choice can be revealed only if all remaining voters in a precinct (and the authorities) collude against him.

At the same time we will mandate that the privacy enhancing mechanism should not disrupt, or interfere in any way with the way the host protocol operates (hence the mechanism should act as a "plug-in" to a host e-voting scheme). For example, if one precinct in a multi-precinct election applies the utility while others do not, it should be possible to employ the protocol without the utility in the other precincts.

In this section we describe how non-interactive zero-sharing can be used as a plug-in to the efficient e-voting scheme of Cramer, Gennaro and Schoenmakers [CGS97]. A *generic* description of our method together with an axiomatic treatment of the necessary properties appears in [KY02b].

6.1 Maximal Ballot Secrecy

The basic security property that is satisfied by an instantiation of the voting utility we propose is Maximal Ballot Secrecy: the sub-tally of the votes of a subset of voters A from the set $\{V_1, ..., V_n\}$ should only be accessible to a malicious coalition of the authorities *and* the remaining voters $\{V_1, ..., V_n\} - A$. In particular this means that only the partial-tally of all the votes of $\{V_1, ..., V_n\}$ will be accessible to the authorities. On the other hand, the violation of the privacy of a

Prover (V_j)		Verifier
publishes $\langle A, B\rangle := \langle g^r, h^r (R'_j)^{\alpha_j^{-1}} f_i\rangle$		
for $\ell \in \{1, \ldots, c\} - \{i\}, d_\ell \in_R \mathbf{Z}_q$		
for $\ell = 1, \ldots, c$, $r_\ell, e_\ell \in_R \mathbf{Z}_q, w, w' \in_R \mathbf{Z}_q$		
$a_\ell = A^{-d_\ell} g^{r_\ell}$		
$a_i = g^w$		
$b_\ell = (B/f_\ell)^{-d_\ell} h^{r_\ell} (R'_j)^{e_\ell}$		
$b_i = h^w (R'_j)^{w'}$		
$c_\ell = h^{-d_\ell} (R'_j)^{e_\ell}$		
$c_i = (h_j)^{w'}$		
	$\{a_\ell\}_\ell, \{b_\ell\}_\ell, \{c_\ell\}_\ell$ $\xrightarrow{\quad}$	
	$\xleftarrow[\quad]{c}$	$c \in_R \mathbf{Z}_q$
$d_i := c - (\sum_{\ell \neq i} d_\ell)$		
$r_i := w + r d_i$		
$e_i := w' + \alpha_j^{-1} d_i$	$\{r_\ell\}_\ell, \{d_\ell\}_\ell, \{e_\ell\}_\ell$ $\xrightarrow{\quad}$	$c \stackrel{?}{=} \sum_\ell d_\ell$
		for $\ell = 1, \ldots, c$,
		$g^{r_\ell} \stackrel{?}{=} a_\ell A^{d_\ell}$
		$h^{r_\ell} (R'_j)^{e_\ell} \stackrel{?}{=} b_\ell (B/f_\ell)^{d_\ell}$
		$(h_j)^{e_\ell} \stackrel{?}{=} c_\ell h^{d_\ell}$

Fig. 3. The interactive version of the proof of ballot validity

single voter V_i requires the collaboration of the authorities with all other voters in the set $\{V_1, V_2, \ldots, V_{i-1}, V_{i+1}, \ldots V_n\}$.

We remark that our description of the security property applies to the state of knowledge prior to the announcement of the final tally. After the announcement of the final tally the knowledge of every participant about what others voted may increase dramatically (e.g. in a yes/no voting procedure between 10 participants that ends up with 10 "yes" and 0 "no" votes there is no question after the announcement of the final tally what each individual voter voted). As a result the security of the voters is understood to be conditioned by the uncertainty that remains after the tally announcement. A formal security treatment of maximal ballot secrecy will be provided in the full version.

6.2 Description of the [CGS97] scheme

We give a brief description of the [CGS97]-scheme, for more details the reader is referred to the original paper. The protocol is initialized by a small set of authorities $A_1, \ldots, A_m$ that setup a public-key for threshold ElGamal encryption. In particular a public-key $g, h \in G$ is published, with the property that any m' of the authorities, given $\langle A, B\rangle$ can compute in a distributed publicly verifiable manner the value $B(A^{\log_g h})^{-1}$ (the ElGamal decryption of the ciphertext $\langle A, B\rangle$); for more details on such threshold schemes see e.g. [GJKR99]. Additionally the generators $f_1, \ldots, f_c$ of G with unknown relative discrete-logs become known to all participants.

Ballot-casting is performed using a bulletin board, where each user publishes his encrypted vote $C := \langle g^r, h^r f_v \rangle$, where $v \in \{1, \ldots, c\}$ is the choice of the voter, and r is selected at random from $\mathbf{Z}_q$. The voter also writes a non-interactive proof of knowledge that ensures that C is a valid ElGamal encryption of one of the values $f_1, \ldots, f_c$ (this can be easily implemented as a proof of knowledge of a valid ElGamal ciphertext combined with a standard OR-argument).

After the ballot-casting procedure terminates, the bulletin board server aggregates all votes by multiplying them point-wise, and due to the homomorphic property of ElGamal encryption, the result $\langle A, B \rangle$ is a valid ElGamal encryption of the plaintext $f_1^{C_1} \ldots f_c^{C_c}$ where C_i denotes the number of votes that the candidate i received. The authorities $A_1, \ldots, A_m$ collaboratively invert the ciphertext and the value $f_1^{C_1} \ldots f_c^{C_c}$ is revealed. Subsequently, using a brute-force search in the set $\{f_1^{x_1} \ldots f_c^{x_c} \mid x_1, \ldots, x_c\}$ the exact counts are revealed (note that the [CGS97]-scheme is efficient only in the case where the number of candidates is considered to be small — logarithmic in the security parameter)

Observe that the privacy of the ballots relies to the assumption that the number of dishonest authorities among $A_1, \ldots, A_m$ is at most $m' - 1$.

6.3 The Enhanced Ballot-Casting Procedure

Suppose that a subset of voters $\{V_1, \ldots, V_n\}$ from the voter population wants to achieve maximal ballot privacy (beyond the conditional authority-based privacy allowed by the [CGS97]-scheme). Prior to ballot-casting the set of voters $V_1, \ldots, V_n$ executes the non-interactive zero-sharing tool of section 3.4. Subsequently each voter V_j that belongs in this group modifies the ballot-casting procedure, by publishing as its encrypted ballot the following tuple: $\langle g^r, h^r (R'_j)^{\alpha_j^{-1}} f_i \rangle$.

Now observe that if $\langle A_1, B_1 \rangle, \ldots, \langle A_n, B_n \rangle$ are the encrypted ballots of the users in the group it holds that their point-wise multiplication $\langle A_1 \ldots A_n, B_1 \ldots B_n \rangle$ is a valid ElGamal encryption of their votes' partial-sum (because of the cancellation property of the zero-shares). On the other hand any set of malicious authorities (even above the threshold m') are incapable of decrypting any $\langle A_i, B_i \rangle$ as the vote of the i-th voter is "blinded" (since it is multiplied by the exponentiated zero-share h^{t_i}). In the full version the following proposition will be shown, that follows from fact 1:

Proposition 5. *The privacy enhancing voting utility supports maximal ballot secrecy, under the DDH assumption.*

Finally, we show in figure 3 how the proof of ballot validity needs to be modified so that voters demonstrate they have blinded their encrypted ballot properly (this allows a seamless integration of the utility into the [CGS97]-scheme).

References

[Bea97] Donald Beaver, *Commodity-Based Cryptography*, STOC 1997.
[Ben87] Josh Benaloh, *Verifiable Secret-Ballot Elections*, PhD Thesis, Yale U., 1987.

[BY86] Josh Benaloh and Moti Yung, *Distributing the Power of a Government to Enhance the Privacy of Voters*, PODC 1986.

[BT94] Josh Benaloh and Dwight Tuinstra, *Receipt-Free Secret-Ballot Elections*, STOC 1994.

[Bra99] Stefan Brands, *Rethinking Privacy*, Ph.D. thesis, pages 230-231.

[Cha81] David Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM 24(2): 84-88, 1981.

[Cha88] David Chaum, *Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA*, EUROCRYPT 1988.

[CP93] David Chaum and Torben P. Pedersen, *Wallet Databases with Observers*, CRYPTO 1992.

[CF85] Josh D. Cohen (Benaloh) and Michael J. Fischer, *A Robust and Verifiable Cryptographically Secure Election Scheme*, FOCS 1985.

[CGS97] Ronald Cramer, Rosario Gennaro and Berry Schoenmakers, *A Secure and Optimally Efficient Multi-Authority Election Scheme*, EUROCRYPT 1997.

[CDS94] Ronald Cramer, Ivan Damgård and Berry Schoenmakers, *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols*, CRYPTO 1994.

[CFSY96] Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers and Moti Yung, *Multi-Autority Secret-Ballot Elections with Linear Work*, EUROCRYPT 1996.

[DDPY94] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, Moti Yung, *On Monotone Formula Closure of SZK*, FOCS 1994.

[FS87] Amos Fiat and Adi Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, CRYPTO 1986.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto and Kazuo Ohta: *A Practical Secret Voting Scheme for Large Scale Elections*, ASIACRYPT 1992.

[GMW87] Oded Goldreich, Silvio Micali, Avi Wigderson, *How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority*, STOC 1987.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, *Secure Distributed Key Generation for Discrete-Log Based Cryptosystems*, EUROCRYPT 1999.

[HS00] Martin Hirt and Kazue Sako, *Efficient Receipt-Free Voting Based on Homomorphic Encryption*, EUROCRYPT 2000.

[KY02] Aggelos Kiayias and Moti Yung, *Self-Tallying Elections and Perfect Ballot Secrecy*, Public-Key Cryptography 2002.

[KY02b] Aggelos Kiayias and Moti Yung, *Robust verifiable non-interactive zero-sharing: A plug-in utility for enhanced voters' privacy*, Chapter 9, **Secure Electronic Voting**, Ed. D. Gritzalis, Advances in Information Security, Vol. 7, Kluwer Academic Publishers, Boston, 2002, pp. 139-152.

[SK94] Kazue Sako and Joe Kilian, *Secure Voting Using Partially Compatible Homomorphisms,* CRYPTO 1994.

[Sch99] Berry Schoenmakers, *A Simple Publicly Verifiable Secret Sharing Scheme and its Applications to Electronic Voting*, CRYPTO 1999.

Author Index

Acquisti, Alessandro 84

Blömer, Johannes 162
Brandt, Felix 223
Buttyán, Levente 15

Dingledine, Roger 84

Foley, Simon N. 1

Garay, Juan A. 190
Gaud, Matthieu 34
Goldie-Scot, Duncan 69

Herranz, Javier 286
Hubaux, Jean-Pierre 15

Jakobsson, Markus 15
Jones, Tim 69
Juels, Ari 103

Kiayias, Aggelos 303
Kügler, Dennis 149
Kuhlmann, Dirk 255

Odlyzko, Andrew 69, 77, 182

Pappu, Ravikanth 103
Pomerance, Carl 190

Rivest, Ron 69

Sáez, Germán 286
Schechter, Stuart E. 122
Seifert, Jean-Pierre 162
Sella, Yaron 270
Smith, Michael D. 122
Someren, Nicko van 69
Stern, Jacques 138
Stern, Julien P. 138
Suzuki, Koutarou 239
Syverson, Paul 84

Traoré, Jacques 34

Vogt, Holger 208

Xu, Shouhuai 51

Yokoo, Makoto 239
Yung, Moti 51, 250, 303