Simulatable Adaptive Oblivious Transfer

Jan Camenisch[1], Gregory Neven[2,3], and abhi shelat[1]

[1] IBM Research, Zurich Research Laboratory, CH-8803 Rüschlikon
[2] Katholieke Universiteit Leuven, Dept. of Electrical Engineering, B-3001 Heverlee
[3] Ecole Normale Supérieure, Département d'Informatique, 75230 Paris Cedex 05

Abstract. We study an *adaptive* variant of oblivious transfer in which a sender has N messages, of which a receiver can adaptively choose to receive k one-after-the-other, in such a way that (a) the sender learns nothing about the receiver's selections, and (b) the receiver only learns about the k requested messages. We propose two practical protocols for this primitive that achieve a stronger security notion than previous schemes with comparable efficiency. In particular, by requiring full simulatability for both sender and receiver security, our notion prohibits a subtle selective-failure attack not addressed by the security notions achieved by previous practical schemes.

Our first protocol is a very efficient generic construction from unique blind signatures in the random oracle model. The second construction does not assume random oracles, but achieves remarkable efficiency with only a constant number of group elements sent during each transfer. This second construction uses novel techniques for building efficient simulatable protocols.

1 Introduction

The *oblivious transfer* (OT) primitive, introduced by Rabin [Rab81], and extended by Even, Goldreich, and Lempel [EGL85] and Brassard, Crépeau and Robert [BCR87] is deceptively simple: there is a sender S with messages $M_1, \ldots, M_N$ and a receiver R with a selection value $\sigma \in \{1, \ldots, N\}$. The receiver wishes to retrieve M_σ from S in such a way that (1) the sender does not "learn" anything about the receiver's choice σ and (2) the receiver "learns" only M_σ and nothing about any other message M_i for $i \neq \sigma$. Part of the allure of OT is that it is *complete*, i.e., if OT can be realized, virtually any secure multiparty computation can be [GMW87,CK90].

In this paper, we consider an *adaptive* version of oblivious transfer in which the sender and receiver first run an initialization phase during which the sender commits to a "database" containing her messages. Later on, the sender and receiver interact as before so that the receiver can retrieve some message M_σ. In addition, we allow the receiver to interact with the sender $k-1$ additional times, one interaction after the other, in order to retrieve additional values from the sender's database. Notice here that we specifically model the situation in which the receiver's selection in the ith phase can *depend* on the messages retrieved in the prior $i-1$ phases. This type of adaptive OT problem is central to a variety of practical problems such as patent searches, treasure hunting, location-based services, oblivious search, and medical databases [NP99b].

The practicality of this adaptive OT problem also drives the need for efficient solutions to it. Ideally, a protocol should only require communication linear in N and the security parameter κ during the initialization phase (so that the sender commits to the N messages), and an amount of communication of $O(\max(\kappa, \log N))$ during each transfer phase (so that the receiver can use cryptography and encode the index of his choice).[4] In the race to achieve these efficiency parameters, however, we must also not overlook—or worse, *settle* for less-than-ideal security properties.

1.1 Security Definitions of Oblivious Transfer

An important contribution of this work is that it achieves a stronger simulation-based security notion at very little cost with respect to existing schemes that achieve weaker notions. We briefly summarize the various security notions for OT presented in the literature, and how our notion extends them.

Honest-but-curious model. In this model, all parties are assumed to follow the protocol honestly. Security guarantees that after the protocol completes, a curious participant cannot analyze the transcript of the protocol to learn anything else. Any protocol in the honest-but-curious model can be transformed into fully-simulatable protocols, albeit at the cost of adding complexity assumptions and requiring costly general zero-knowledge proofs for each protocol step.

Half-simulation. This notion, introduced by Naor and Pinkas [NP05], considers malicious senders and receivers, but handles their security separately. Receiver security is defined by requiring that the sender's view of the protocol when the receiver chooses index σ is indistinguishable from a view of the protocol when the receiver chooses σ'. Sender security, on the other hand, involves a stronger notion. The requirement follows the real-world/ideal-world paradigm and guarantees that any malicious receiver in the real world can be mapped to a receiver in an idealized game in which the OT is implemented by a trusted party. Usually, this requires that receivers are efficiently "simulatable," thus we refer to this notion as *half-simulation.*

The Problem of Selective Failure. We argue that the definition of half-simulation described above does not imply all properties that one may expect from a $\mathcal{OT}_{k\times 1}^{N}$ scheme. Notice that a cheating sender can always make the current transfer fail by sending bogus messages. However, we would not expect him to

[4] In practice, we assume that $\kappa > \log(N)$—so that the protocol can encode the receiver's selection—but otherwise that κ is chosen purely for the sake of security. In this sense, $O(\kappa)$ is both conceptually and practically different than $O(\text{polylog}(N))$.

be able to cause failure based on some property of the receiver's selection. Of course, the sener can also prevent the receiver from retrieving M_σ by replacing it with a random value during the initialization phase. But again, the sender should not be able to make this decision anew at each transfer phase. For example, the sender should not be able to make the first transfer fail for $\sigma = 1$ but succeed for $\sigma \in \{2, \ldots, N\}$, and to make the second transfer fail for $\sigma = 2$ but succeed for $\sigma \in \{1, 3, \ldots, N\}$. The receiver could publicly complain whenever a transfer fails, but by doing so it gives up the privacy of its query. Causing transfers to fail may on the long term harm the sender's business, but relying on such arguments to dismiss the problem is terribly naive. A desperate patent search database may *choose* to make faster money by selling a company's recent queries to competitors than by continuing to run its service.

We refer to this issue as the *selective-failure* problem. To see why it is not covered by the half-simulation notion described above, it suffices to observe that the notion of receiver security only *hides* the message received by the receiver from the cheating sender's view. A scheme that is vulnerable to selective-failure attacks does not give the cheating sender any additional advantage in breaking the receiver's privacy, and may therefore be secure under such a notion. (This illustrates the classic argument from work in secure multiparty computation that achieving just privacy is not enough; both privacy and correctness must be achieved simultaneously.) In fact, the schemes of [NP05] are secure under half-simulation, yet vulnerable to selective-failure attacks. In an earlier version [NP99b], the same authors recognize this problem and remark that it can be fixed, but do not give formal support of their claim. A main contribution of this work is to show that it can be done without major sacrifices in efficiency.

Simulatable OT. The security notion that we consider employs the real-world/ideal-world paradigm for both receiver and sender security. We extend the functionality of the trusted party such that at each transfer, the sender inputs a bit b indicating whether it wants the transfer to succeed or fail. This models the capability of a sender in the real world to make the transfer fail by sending bogus messages, but does not enable it to do so based on the receiver's input σ. Moreover, for security we require indistinguishability of the combined outputs of the sender and the receiver, rather than only of the output of the dishonest party. The output of the honest receiver is assumed to consist of all the messages $M_{\sigma_1}, \ldots, M_{\sigma_k}$ that it received. This security notion excludes selective-failure attacks in the real world, because the ideal-world sender is unable to perform such attacks, which will lead to noticeable differences in the receiver's output in the real and ideal world.

Finally, we observe that simulatable oblivious transfer is used as a primitive to build many other cryptographic protocols [Gol04]. By building an efficient OT protocol with such simulation, we take the first steps at realizing many other interesting cryptographic protocols.

1.2 Construction Overview

Our random-oracle protocol. Our first construction is a black-box construction using any unique blind signature scheme. By *unique*, we mean that for all public keys and messages there exists at most one valid signature. First, the sender generates a key pair (pk, sk) for the blind signature scheme, and "commits" to each message in its database by XOR-ing the message M_i with $H(i, s_i)$, where s_i is the unique signature of the message i under pk. Intuitively, we're using s_i as a key to unlock the message M_i. To retrieve the "key" to a message M_i, the sender and receiver engage in the blind signature protocol for message i. By the unforgeability of the signature scheme, a malicious receiver will be unable to unlock more than k such messages. By the blindness of the scheme, the sender learns nothing about which messages have been requested.

The random oracle serves four purposes. First, it serves as a one-time pad to perfectly hide the messages. Second, it allows a simulator to extract the sender's original messages from the commitments so that we can prove receiver-security. Third, in the proof of sender-security, it allows the simulator to both extract the receiver's choice and, via programming the random oracle, to make the receiver open the commitment to an arbitrary message. Finally, it allows us to extract forgeries of the blind signature scheme from a malicious receiver who is able to break sender-security.

Our standard-model protocol. There are three main ideas behind the standard protocol in §4. At a very high level, just as in the random oracle protocol, the sender uses a unique signature of i as a key to encrypt M_i in the initialization phase. However, unlike the random-oracle protocol, we observe here that we only need a blind signature scheme which allows signatures on a small, *a-priori fixed* message space $\{1, \dots, N\}$.

The second idea concerns the fact that after engaging in the blind-signing protocol, a receiver can easily check whether the sender has sent the correct response during the transfer phase by verifying the signature it received. While seemingly a feature, this property becomes a problem during the simulation of a malicious receiver. Namely, the simulator must commit to N random values during the initialize phase, and later during the transfer phase, open any one of these values to an arbitrary value (the correct message M_i received from the trusted party during simulation). In the random oracle model, this is possible via programming the random oracle. In the standard model, a typical solution would be to use a trapdoor commitment. However, a standard trapdoor commitment is unlikely to work here because most of these require the opener to send the actual committed value when it opens the commitment. This is not possible in our OT setting since the sender does not know which commitment is being opened.

Our solution is to modify the "blind-signing" protocol so that, instead of returning a signature to the user, a one-way function (a bilinear pairing in our case) of the signature is returned. To protect against a malicious sender, the sender then proves in zero-knowledge that the value returned is computed correctly. In the security proof, we will return a random value to the receiver and fake the zero-knowledge proof.

The final idea behind our construction concerns a malicious receiver who may use an invalid input to the "blind-signature protocol" in order to, say, retrieve a signature on a value outside of $\{1, \ldots, N\}$. This is a real concern, since such an attack potentially allows a malicious receiver to learn the product $M_i \cdot M_j$ which violates the security notion. In order to prevent such cheating, we require the receiver to prove in zero-knowledge that (a) it knows the input it is requesting a signature for, and (b) that the input is valid for the protocol. While this is conceptually simple, the problem is that the size of such a theorem statement, and therefore the time and communication complexity of such a zero-knowledge proof, could potentially be linear in N. For our stated efficiency goals, we need a proof of constant size. To solve this final problem, we observe that the input to the blind signature process is a small set—i.e., only has N possible values. Thus, the sender can sign all N possible input messages (using a different signing key x) to the blind signature protocol and publish them in the initialization phase. During the transfer phase, the receiver blinds one of these inputs and then gives a zero-knowledge proof of knowledge that it knows a signature of this blinded input value. Following the work of Camenisch and Lysyanskaya [CL04], there are very efficient proofs for such statements which are constant size.

Finally, in order to support receiver security, the sender provides a proof of knowledge of the "commitment key" used to commit to its input message. This key can thus be extracted from the proof of knowledge and use it to compute messages to send to the trusted party.

1.3 Related Work

The concept of oblivious transfer was proposed by Rabin [Rab81] (but considered earlier by Wiesner [Wie83]) and further generalized to one-out-of-two OT ($\mathcal{OT}_1^2$) by Even, Goldreich and Lempel [EGL85] and one-out-of-N OT ($\mathcal{OT}_1^N$) by Brassard, Crépeau and Robert [BCR87]. A complete history of the work on OT is beyond our scope. In particular, here we do not mention constructions of OT which are based on generic zero-knowlege techniques or setup assumptions. See Goldreich [Gol04] for more details.

Bellare and Micali [BM90] presented practical implementations of $\mathcal{OT}_1^2$ under the honest-but-curious notion and later Naor and Pinkas [NP01] did the same under the half-simulation definition. Brassard et al. [BCR87] showed how to implement $\mathcal{OT}_1^N$ using N applications of a $\mathcal{OT}_1^2$ protocol. Under half-simulation, Naor and Pinkas [NP99a] gave a more efficient construction requiring only $\log N$ $\mathcal{OT}_1^2$ executions. Several direct 2-message $\mathcal{OT}_1^N$ protocols (also under half-simulation) have been proposed in various works [NP01,AIR01,Kal05].

The first adaptive k-out-of-N oblivious transfer ($\mathcal{OT}_{k\times 1}^N$) protocol is due to Naor and Pinkas [NP99b]. Their scheme is secure under half-simulation and involves $O(\log N)$ invocations of a $\mathcal{OT}_1^2$ protocol during the transfer stage. Using optimistic parameters, this translates into a protocol with $O(\log N)$ rounds and at least $O(k \log N)$ communication complexity during the transfer phase. The same authors also propose a protocol requiring 2 invocations of a $\mathcal{OT}_1^{\sqrt{N}}$ protocol. Laur and Lipmaa [LL06] build an $\mathcal{OT}_{k\times 1}^N$ in which k must be a constant. Their security notion specifically *tolerates* selective-failure, and the efficiency of their construction depends on the efficiency of the fully-simulatable $\mathcal{OT}_1^N$ and the equivocable (i.e., trapdoor) list commitment scheme which are used as primitives.

In the random oracle model, Ogata and Kurosawa [OK04] and Chu and Tzeng [CT05] propose two efficient $\mathcal{OT}_{k\times 1}^N$ schemes satisfying half-simulation which require $O(k)$ computation and communication during the transfer stage. Our first generic $\mathcal{OT}_{k\times 1}^N$ construction based on unique blind signatures covers both schemes as special cases, offers full simulation-security, and fixes minor technical problems to prevent certain attacks. Prior to our work, Malkhi and Sella [MS03] observed a relation between OT and blind signatures, but did not give a generic transformation between the two. They present a direct $\mathcal{OT}_1^N$ protocol (also in the random oracle model) based on Chaum's blind signatures [Cha88]. Their scheme could be seen as a $\mathcal{OT}_{k\times 1}^N$ protocol as well, but it has communication complexity $O(\kappa N)$ in the transfer phase. Their scheme is not an instantiation of our generic construction.

$\mathcal{OT}_{k\times 1}^N$ can always be achieved by publishing commitments to the N data items, and executing k $\mathcal{OT}_1^N$ protocols on the N pieces of opening information. This solution incurs costs of $O(\kappa N)$ in each transfer phase.

Naor and Pinkas [NP05] demonstrate a way to transform a singe-server private-information retrieval scheme (PIR) into an oblivious transfer scheme with sublinear-in-N communication complexity. This transformation is in the half-simulation model and the dozen or so constructions of OT from PIR seem to also be in this model. Moreover, there are no adaptive PIR schemes known.

2 Definitions

If $k \in \mathbb{N}$, then 1^k is the string consisting of k ones. The empty string is denoted ε. If A is a randomized algorithm, then $y \xleftarrow{\$} \mathsf{A}(x)$ denotes the assignment to y of the output of A on input x when run with fresh random coins. Unless noted, all algorithms are probabilistic polynomial-time (PPT) and we implicitly assume they take an extra parameter 1^κ. A function $\nu : \mathbb{N} \to [0,1]$ is *negligible* if for all $c \in \mathbb{N}$ there exists a $\kappa_c \in \mathbb{N}$ such that $\nu(\kappa) < \kappa^{-c}$ for all $\kappa > \kappa_c$.

2.1 Blind Signatures

A blind signature scheme $\mathcal{BS}$ is a tuple of PPT algorithms $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{User}, \mathsf{Vf})$. The signer generates a key pair via the key generation algorithm $(pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\kappa)$. To obtain a signature on a message M, the user and signer engage in an interactive signing protocol dictated by the $\mathsf{User}(pk, M)$ and $\mathsf{Sign}(sk)$ algorithms.

At the end of the protocol, the User algorithm returns a signature s or $\perp$ to indicate rejection. The verification algorithm $\mathsf{Vf}(pk, M, s)$ returns 1 if the signature is deemed valid and 0 otherwise. Correctness requires that $\mathsf{Vf}(pk, M, s) = 1$ for all (pk, sk) output by the Kg algorithm, for all $M \in \{0,1\}^*$ and for all signatures output by $\mathsf{User}(pk, M)$ after interacting with $\mathsf{Sign}(sk)$. We say that $\mathcal{BS}$ is *unique* [GO92] if for each public key $pk \in \{0,1\}^*$ and each message $M \in \{0,1\}^*$ there exists at most one signature $s \in \{0,1\}^*$ such that $\mathsf{Vf}(pk, M, s) = 1$.

The security of blind signatures is twofold. On the one hand, *one-more unforgeability* [PS96] requires that no adversary can output $n+1$ valid message-signature pairs after being given the public key as input and after at most n interactions with a signing oracle. We say that $\mathcal{BS}$ is unforgeable if no PPT adversary has non-negligible probability of winning this game.

Blindness, on the other hand, requires that the signer cannot tell apart the message it is signing. The notion was first formalized by Juels et al. [JLO97], and was later strengthened to *dishonest-key blindness* [ANN06,Oka06]. In this work, we further strengthen the definition to *selective-failure blindness*. It is defined through the following game. The adversary first outputs a public key pk and two messages M_0, M_1. It is then given black-box access to two instances of the user algorithm, the first implementing $\mathsf{User}(pk, M_b)$ and the second implementing $\mathsf{User}(pk, M_{1-b})$ for a random bit $b \stackrel{\$}{\leftarrow} \{0,1\}$. Eventually, these algorithms produce local output s_b and s_{1-b}, respectively. If $s_b \neq \perp$ and $s_{1-b} \neq \perp$, then the adversary is given the pair (s_0, s_1); if $s_b = \perp$ and $s_{1-b} \neq \perp$, then it is given $(\perp, \varepsilon)$; if $s_b \neq \perp$ and $s_{1-b} = \perp$, then it is given $(\varepsilon, \perp)$; and if $s_b = s_{1-b} = \perp$ it is given $(\perp, \perp)$. The adversary then guesses the bit b. The scheme $\mathcal{BS}$ is said to be selective-failure blind if no PPT adversary has a non-negligible advantage in winning the above game.

2.2 Simulatable Adaptive Oblivious Transfer

An adaptive k-out-of-N oblivious transfer scheme $\mathcal{OT}_{k\times 1}^{N}$ is a tuple of four PPT algorithms $(\mathsf{S_I}, \mathsf{R_I}, \mathsf{S_T}, \mathsf{R_T})$. During the initialization phase, the sender and receiver perform an interactive protocol where the sender runs the $\mathsf{S_I}$ algorithm on input messages $M_1, \ldots, M_N$, while the receiver runs the $\mathsf{R_I}$ algorithm without input. At the end of the initialization protocol, the $\mathsf{S_I}$ and $\mathsf{R_I}$ algorithm produce as local outputs state information S_0 and R_0, respectively. During the i-th transfer, $1 \leq i \leq k$, the sender and receiver engage in a selection protocol dictated by the $\mathsf{S_T}$ and $\mathsf{R_T}$ algorithms. The sender runs $\mathsf{S_T}(S_{i-1})$ to obtain updated state information S_i, while the receiver runs the $\mathsf{R_T}$ algorithm on input state information R_{i-1} and the index σ_i of the message it wishes to receive, to obtain updated state information R_i and the retrieved message M'_{σ_i}. Correctness requires that $M'_{\sigma_i} = M_{\sigma_i}$ for all messages $M_1, \ldots, M_N$, for all selections $\sigma_1, \ldots, \sigma_k \in \{1, \ldots, N\}$ and for all coin tosses of the algorithms.

To capture security of an $\mathcal{OT}_{k\times 1}^{N}$ scheme, we employ the real-world/ideal-world paradigm. Below, we describe a real experiment in which the parties run the protocol, while in the ideal experiment the functionality is implemented through a trusted third party. For the sake of simplicity, we do not explicitly include auxiliary inputs to the parties. This can be done, and indeed must be done for sequential composition of the primitive, and our protocols achieve this notion as well.

Real experiment. We first explain the experiment for arbitrary sender and receiver algorithms $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$. The experiment $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma)$ proceeds as follows. $\widehat{\mathsf{S}}$ is given messages $(M_1, \ldots, M_N)$ as input and interacts with $\widehat{\mathsf{R}}(\Sigma)$, where Σ is an adaptive selection algorithm that, on input messages $M_{\sigma_1}, \ldots, M_{\sigma_{i-1}}$, outputs the index σ_i of the next message to be queried. In their first run, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ produce initial states S_0 and R_0 respectively. Next, the sender and receiver engage in k interactions. In the i-th interaction for $1 \leq i \leq k$, the sender and receiver interact by running $S_i \xleftarrow{\$} \widehat{\mathsf{S}}(S_{i-1})$ and $(R_i, M_i^*) \xleftarrow{\$} \widehat{\mathsf{R}}(R_{i-1})$, and update their states to S_i and R_i, respectively. Note that M_i^* may be different from M_{σ_i} when either participant cheats. At the end of the k-th interaction, sender and receiver output strings S_k and R_k respectively. The output of the $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}$ experiment is the tuple (S_k, R_k).

For an $\mathcal{OT}^N_{k\times 1}$ scheme $(\mathsf{S_I}, \mathsf{S_T}, \mathsf{R_I}, \mathsf{R_T})$, define the honest sender S algorithm as the one which runs $\mathsf{S_I}(M_1, \ldots, M_N)$ in the initialization phase, runs $\mathsf{S_T}$ in all following interactions, and always outputs $S_k = \varepsilon$ as its final output. Define the honest receiver R as the algorithm which runs $\mathsf{R_I}$ in the initialization phase, runs $\mathsf{R_T}(R_{i-1}, \sigma_i)$ and in the i-th interaction, where Σ is used to generate the index σ_i, and returns the list of received messages $R_k = (M_{\sigma_1}, \ldots, M_{\sigma_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$, the (possibly cheating) sender algorithm $\widehat{\mathsf{S}}'(M_1, \ldots, M_N)$ generates messages $M_1^*, \ldots, M_N^*$ and hands these to the trusted party T. In each of the k transfer phases, T receives a bit b_i from the sender $\widehat{\mathsf{S}}'$ and an index σ_i^* from the (possibly cheating) receiver $\widehat{\mathsf{R}}'(\Sigma)$. If $b_i = 1$ and $\sigma_i^* \in \{1, \ldots, N\}$, then T hands $M^*_{\sigma_i^*}$ to the receiver; otherwise, it hands $\perp$ to the receiver. At the end of the k-th transfer, $\widehat{\mathsf{S}}'$ and $\widehat{\mathsf{R}}'$ output a string S_k and R_k; the output of the experiment is the pair (S_k, R_k).

As above, define the ideal sender $\mathsf{S}'(M_1, \ldots, M_N)$ as one who sends messages $M_1, \ldots, M_N$ to the trusted party in the initialization phase, sends $b_i = 1$ in all transfer phases, and uses $S_k = \varepsilon$ as its final output. Define the honest ideal receiver R' as the algorithm which generates its selection indices σ_i through Σ and submits these to the trusted party. Its final output consists of all the messages it received $R_k = (M_{\sigma_i}, \ldots, M_{\sigma_N})$.

Sender security. We say that $\mathcal{OT}^N_{k\times 1}$ is sender-secure if for any PPT real-world cheating receiver $\widehat{\mathsf{R}}$ there exists a PPT ideal-world receiver $\widehat{\mathsf{R}}'$ such that for any polynomial $N_m(\kappa)$, any $N \in [1, N_m(\kappa)]$, any $k \in \{1, \ldots, N\}$, any messages $M_1, \ldots, M_N$, and any selection strategy Σ, the advantage of any PPT distinguisher in distinguishing the distributions

$$\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma) \quad \text{and} \quad \mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$$

is negligible in κ.

Receiver security. We say that $\mathcal{OT}^N_{k\times 1}$ is receiver-secure if for any PPT real-world cheating sender $\widehat{\mathsf{S}}$ there exists a PPT ideal-world sender $\widehat{\mathsf{S}}'$ such that for any polynomial $N_m(\kappa)$, any $N \in [1, N_m(\kappa)]$, any $k \in \{1,\ldots,N\}$, any messages $M_1,\ldots,M_N$, and any selection strategy Σ, the advantage of any PPT distinguisher in distinguishing the distributions

$$\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N,k,M_1,\ldots,M_N,\Sigma) \quad \text{and} \quad \mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N,k,M_1,\ldots,M_N,\Sigma)$$

is negligible in κ.

3 A Generic Construction in the Random Oracle Model

In this section, we describe a generic yet very efficient way of constructing adaptive k-out-of-N OT schemes from unique blind signature schemes, and prove its security in the random oracle model.

3.1 The Construction

To any unique blind signature scheme $\mathcal{BS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{User}, \mathsf{Vf})$, we associate the $\mathcal{OT}^N_{k\times 1}$ scheme as depicted in Fig. 1. The security of the oblivious transfer

INITIALIZATION

$$\begin{array}{lcl}
\underline{\mathsf{S_I}(M_1,\ldots,M_N):} & & \underline{\mathsf{R_I}:} \\
\quad (pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\kappa) & & \\
\quad \text{For } i = 1\ldots N & & \\
\quad\quad s_i \leftarrow \mathsf{Sign}(sk, i) & & \\
\quad\quad C_i \leftarrow H(i, s_i) \oplus M_i & \xrightarrow{pk,\, C_1,\ldots,C_N} & R_0 \leftarrow (pk, C_1,\ldots,C_N) \\
\quad S_0 \leftarrow sk \text{ ; Output } S_0 & & \text{Output } R_0
\end{array}$$

TRANSFER

$$\begin{array}{lcl}
\underline{\mathsf{S_T}(S_{i-1}):} & & \underline{\mathsf{R_T}(R_{i-1}, \sigma_i):} \\
\quad \text{Parse } S_{i-1} \text{ as } sk & & \text{Parse } R_{i-1} \text{ as } (pk, C_1,\ldots,C_N) \\
\quad \text{Run protocol } \mathsf{Sign}(sk) & \longleftrightarrow & \text{Run protocol } s_i \xleftarrow{\$} \mathsf{User}(pk, \sigma_i) \\
 & & \text{If } \mathsf{Vf}(pk, \sigma_i, s_i) = 0 \text{ then } M_{\sigma_i} \leftarrow \bot \\
 & & \text{Else } M_{\sigma_i} \leftarrow C_{\sigma_i} \oplus H(i, s_i) \\
\quad \text{Output } S_i = S_{i-1} & & \text{Output } (R_i = R_{i-1}, M_{\sigma_i})
\end{array}$$

Fig. 1. A construction of $\mathcal{OT}^N_{k\times 1}$ using a random oracle H and any unique blind signature scheme $\mathcal{BS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{User}, \mathsf{Vf})$.

scheme follows from that of the blind signature scheme. In particular, Theorem 1 states that the sender's security is implied by the one-more unforgeability of $\mathcal{BS}$, while Theorem 2 states that the receiver's security follows from the selective-failure blindness of $\mathcal{BS}$. We provide brief proof sketches below; detailed proofs can be found in the full version [CNS07].

Theorem 1. *If the blind signature scheme $\mathcal{BS}$ is unforgeable, then the $\mathcal{OT}^N_{k\times 1}$ depicted in Fig. 1 is sender-secure in the random oracle model.*

Proof (Sketch). The idea of this proof is that the ideal-world receiver $\widehat{\mathsf{R}}'$ runs the real-world receiver $\widehat{\mathsf{R}}$ on random ciphertexts $C_1, \ldots, C_N$. $\widehat{\mathsf{R}}'$ observes the random oracle queries made by $\widehat{\mathsf{R}}$, trying to parse them as $H(\sigma_i, s_i)$ such that $\mathsf{Vf}(pk, \sigma_i, s_i)$=1. If this succeeds, then it requests M_{σ_i} from the trusted party and returns $C_{\sigma_i} \oplus M_{\sigma_i}$. If there are more than k such queries, then $\widehat{\mathsf{R}}$ has forged the blind signature scheme.

Theorem 2. *If the blind signature scheme $\mathcal{BS}$ is selective-failure blind, then the $\mathcal{OT}^N_{k\times 1}$ scheme depicted in Fig. 1 is receiver-secure in the random oracle model.*

Proof (Sketch). For any real-world cheating sender $\widehat{\mathsf{S}}$, consider the ideal-world cheating sender $\widehat{\mathsf{S}}'$ that, when $\widehat{\mathsf{S}}$ outputs $(pk, C_1, \ldots, C_N)$, goes over all random oracle queries made by $\widehat{\mathsf{S}}$ and tries to parse them as (i, s_i) such that $\mathsf{Vf}(pk, i, s_i) = 1$. If this succeeds it sets $M_i \leftarrow C_i \oplus H(i, s_i)$, all other messages are chosen at random. It then submits $M_1, \ldots, M_N$ to the trusted party. At the i-th transfer, $\widehat{\mathsf{S}}'$ runs $\widehat{\mathsf{S}}$ against $(R_i, M_i^*) \xleftarrow{\$} \mathsf{R}_\mathsf{T}(R_{i-1}, 1)$. If $M_i^* = \perp$ then $\widehat{\mathsf{S}}'$ submits a zero bit to the trusted party, indicating that the present transfer should fail, otherwise it sends a one. The selective-failure blindness of $\mathcal{BS}$ ensures that $\widehat{\mathsf{S}}$ cannot distinguish a query for index 1 from any other query, and that it cannot make R_T fail depending on the value of its selection.

Instantiations. Many blind signature schemes exist, but only the schemes of Chaum [Cha88,BNPS03] and Boldyreva [Bol03] seem to be unique. Both are efficient two-round schemes which result in round-optimal adaptive oblivious transfer protocols.

The instantiation of our generic construction with Chaum's blind signature scheme coincides with the direct OT scheme of Ogata-Kurosawa [OK04]. However, special precautions must be taken to ensure that Chaum's scheme is selective-failure blind. For example, the sender must use a prime exponent e greater than the modulus n [ANN06], or must provide a non-interactive proof that $\gcd(e, n) = 1$ [CPP07]. Anna Lysyanskaya suggests having the receiver send e to the sender. This solution is much more efficient than the previous two, but would require re-proving the security of the $\mathcal{OT}^N_{k\times 1}$ scheme since it is no longer an instance of our generic construction. In any case, the authors of [OK04] overlooked this need, which creates the possibility for attacks on the receiver's security of their protocol. For example, a cheating sender could choose $e = 2$ and distinguish between transfers for σ_i and σ_i' for which $\mathrm{H}(\sigma_i)$ is a square modulo n and $\mathrm{H}(\sigma_i')$ is not.

When instantiated with Boldyreva's blind signature scheme [Bol03] based on pairings, our generic construction coincides with the direct OT scheme of Chu-Tzeng [CT05]. A similar issue concerning the dishonest-key blindness of the scheme arises here, but was also overlooked. The sender could for example choose the group to be of non-prime order and break the receiver's security in a similar way as demonstrated above for the scheme of [OK04]. One can strengthen Boldyreva's blind signature scheme to provide selective-failure blindness by letting the user algorithm check that the group is of prime order p and that the generator is of full order p.

4 Simulatable Adaptive OT in the Standard Model

Computational assumptions. Our protocol presented in this section requires bilinear groups and associated hardness assumptions. Let Pg be a pairing group generator that on input 1^κ outputs descriptions of multiplicative groups $\mathbb{G}_1, \mathbb{G}_\mathrm{T}$ of prime order p where $|p| = \kappa$. Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$ and let $g \in \mathbb{G}_1^*$. The generated groups are such that there exists an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathrm{T}$, meaning that (1) for all $a, b \in \mathbb{Z}_p$ it holds that $e(g^a, g^b) = e(g, g)^{ab}$; (2) $e(g, g) \neq 1$; and (3) the bilinear map is efficiently computable.

Definition 3 (Strong Diffie-Hellman Assumption [BB04]). *We say that the ℓ-SDH assumption associated to a pairing generator Pg holds if for all PPT adversaries A, the probability that $A(g, g^x, \ldots, g^{x^\ell})$ where $(\mathbb{G}_1, \mathbb{G}_\mathrm{T}) \xleftarrow{\$} \mathsf{Pg}(1^\kappa)$, $g \xleftarrow{\$} \mathbb{G}_1^*$ and $x \xleftarrow{\$} \mathbb{Z}_p$, outputs a pair $(c, g^{1/(x+c)})$ where $c \in \mathbb{Z}_p$ in negligible in κ.*

Definition 4 (Power Decisional Diffie-Hellman Assumption). *We say that the ℓ-PPDH assumption associated to Pg holds if for all PPT adversaries A, the probability that A on input $(g, g^x, g^{x^2}, \ldots, g^{x^\ell}, H)$ where $(\mathbb{G}_1, \mathbb{G}_\mathrm{T}) \xleftarrow{\$} \mathsf{Pg}(1^\kappa)$, $g \xleftarrow{\$} \mathbb{G}_1^*$, $x \xleftarrow{\$} \mathbb{Z}_p$, $H \xleftarrow{\$} \mathbb{G}_\mathrm{T}$, distinguishes the vector $T = (H^x, H^{x^2}, \ldots, H^{x^\ell})$ from a random vector $T \xleftarrow{\$} \mathbb{G}_\mathrm{T}^\ell$ is negligible in κ.*

Boneh-Boyen signatures. We modify the weakly-secure signature scheme of Boneh and Boyen [BB04] as follows. The scheme uses a pairing generator Pg as defined above. The signer's secret key is $x \xleftarrow{\$} \mathbb{Z}_p$, the corresponding public key is $y = g^x$. The signature on a message M is $s \leftarrow g^{1/(x+M)}$; verification is done by checking that $e(s, y \cdot g^M) = e(g, g)$. This scheme is similar to the Dodis and Yampolskiy verifiable random function [DY05].

Security under weak chosen-message attack is defined through the following game. The adversary begins by outputting ℓ messages $M_1, \ldots, M_\ell$. The challenger generates a fresh key pair and gives the public key to the adversary, together with signatures $s_1, \ldots, s_\ell$ on $M_1, \ldots, M_\ell$. The adversary wins if it succeeds in outputing a valid signature s on a message $M \notin \{M_1, \ldots, M_\ell\}$. The scheme is said to be unforgeable under weak chosen-message attack if no PPT adversary A has non-negligible probability of winning this game. An easy adaptation of the proof of [BB04] can be used to show that this scheme is unforgeable under weak chosen-message attack if the $(\ell + 1)$-SDH assumption holds.

INITIALIZATION

$\underline{\mathsf{S_I}(1^\kappa, M_1, \ldots, M_N):}$ $\qquad\qquad\qquad\qquad\qquad$ $\underline{\mathsf{R_I}(1^\kappa):}$

$(\mathbb{G}_1, \mathbb{G}_\mathrm{T}) \xleftarrow{\$} \mathsf{Pg}(1^\kappa)$

$g, h \xleftarrow{\$} \mathbb{G}_1^* \; ; \; H \leftarrow e(g, h)$

$x \xleftarrow{\$} \mathbb{Z}_p \; ; \; y \leftarrow g^x \; ; \; pk \leftarrow (g, H, y)$

For $i = 1, \ldots, N$ do

$\quad A_i \leftarrow g^{1/(x+i)}$

$\quad B_i \leftarrow e(h, A_i) \cdot M_i$

$\quad C_i \leftarrow (A_i, B_i)$ $\qquad \xrightarrow{\;pk, C_1, \ldots, C_N\;}$

$S_0 \leftarrow (h, pk)$ $\qquad \xrightarrow{\;PoK\{(h):\; H = e(g,h)\}\;}$ $\qquad R_0 \leftarrow (pk, C_1, \ldots, C_N)$

TRANSFER

$\underline{\mathsf{S_T}(S_{i-1}):}$ $\qquad\qquad\qquad\qquad\qquad$ $\underline{\mathsf{R_T}(R_{i-1}, \sigma_i):}$

$\qquad\qquad\qquad\qquad\qquad\qquad$ $v \xleftarrow{\$} \mathbb{Z}_p \; ; \; V \leftarrow (A_{\sigma_i})^v$

$\xleftarrow{\;V\;}$

$\xleftarrow{\;PoK\{(\sigma_i, v) : e(V, y) = e(V, g)^{-\sigma_i} e(g, g)^v\}\;}$

$W \leftarrow e(h, V)$ $\qquad \xrightarrow{\;W\;}$

$\xrightarrow{\;PoM\{(h) : H = e(g,h) \wedge W = e(h, V)\}\;}$

$S_i = S_{i-1}$ $\qquad\qquad\qquad\qquad\qquad$ $M \leftarrow B_{\sigma_i}/(W^{1/v})$

$\qquad\qquad\qquad\qquad\qquad\qquad$ $R_i = R_{i-1}$

Fig. 2. Our $\mathcal{OT}_{k\times 1}^N$ protocol in the standard model associated to pairing generator Pg. We use notation by Camenisch and Stadler [CS97] for the zero-knowledge protocols. They can all be done efficiently (in four rounds and $O(\kappa)$ communication) by using the transformation of [CDM00]. The protocols are given in detail in the full version [CNS07].

Zero-knowledge proofs. We use definitions from [BG92,CDM00]. A pair of interacting algorithms (P, V) is a proof of knowledge (PoK) for a relation $R = \{(\alpha, \beta)\} \subseteq \{0,1\}^* \times \{0,1\}^*$ with knowledge error $\kappa \in [0,1]$ if (1) for all $(\alpha, \beta) \in R$, $\mathsf{V}(\alpha)$ accepts a conversation with $\mathsf{P}(\beta)$ with probability 1; and (2) there exists an expected polynomial-time algorithm E, called the *knowledge extractor*, such that if a cheating prover $\widehat{\mathsf{P}}$ has probability ϵ of convincing V to accept α, then E, when given rewindable black-box access to $\widehat{\mathsf{P}}$, outputs a witness β for α with probability $\epsilon - \kappa$.

A proof system (P, V) is *perfect zero-knowledge* if there exists a PPT algorithm Sim, called the *simulator*, such that for any polynomial-time cheating verifier $\widehat{\mathsf{V}}$ and for any $(\alpha, \beta) \in R$, the outputs of $\widehat{\mathsf{V}}(\alpha)$ after interacting with $\mathsf{P}(\beta)$ and that of $\mathsf{Sim}^{\widehat{\mathsf{V}}(\alpha)}(\alpha)$ are identically distributed.

Our protocol in the standard model is depicted in Fig. 2. All zero-knowledge proofs can be performed efficiently in four rounds and with $O(\kappa)$ communication using the transformation of [CDM00]. The detailed protocols are provided in the full version [CNS07]. We assume that the messages M_i are elements of the target group $\mathbb{G}_T$.[5] The protocol is easily seen to be correct by observing that $W = e(h, A_{\sigma_i})^v$, so therefore $B_{\sigma_i}/W^{1/v} = M_{\sigma_i}$.

We now provide some intuition into the protocol. Each pair (A_i, B_i) can be seen as an ElGamal encryption [ElG85] in $\mathbb{G}_T$ of M_i under public key H. But instead of using random elements from $\mathbb{G}_T$ as the first component, our protocol uses verifiably random [DY05] values $A_i = g^{1/(x+i)}$. It is this verifiability that during the transfer phase allows the sender to check that the receiver is indeed asking for the decryption key for one particular ciphertext, and not for some combination of ciphertexts.

Receiver security. We demonstrate the receiver security of our scheme by proving the stronger property of unconditional statistical indistinguishability. Briefly, the ideal-world sender can extract h from the proof of knowledge in the initialization phase, allowing it to decrypt the messages to send to the trusted party. During the transfer phase, it plays the role of an honest receiver and asks for a randomly selected index. If the real-world sender succeeds in the final proof of membership (PoM) of the well-formedness of W, then the ideal sender sends $b = 1$ to its trusted-party T to indicate continue.

Notice how the sender's response W is simultaneously determined by the initialization phase, unpredictable by the receiver during the transfer phase, but yet *verifiable* once it has been received (albeit, via a zero-knowledge proof). Intuitively, these three properties prevent the selective-failure attack.

Theorem 5. *The $\mathcal{OT}^N_{k\times 1}$ protocol in Fig. 2 is unconditionally receiver-secure.*

Proof. We show that for every real-world cheating sender $\widehat{\mathsf{S}}$ there exists an ideal-world cheating sender $\widehat{\mathsf{S}}'$ such that no distinguisher D, regardless of its running time, has non-negligible probability to distinguish the distributions $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \Sigma)$ and $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \Sigma)$. We do so by considering a sequence of distributions **Game-0**, ..., **Game-3** such that for some $\widehat{\mathsf{S}}'$ that we construct, **Game-0** $=$ $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}$ and **Game-3** $=$ $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}$, and by demonstrating the statistical difference in the distribution for each game transition. Below, we use the shorthand notation

$$\Pr[\,\mathbf{Game\text{-}i}\,] \;=\; \Pr\Big[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Game\text{-}i}\,\Big] \;.$$

Game-0 : This is the distribution corresponding to $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}$, i.e., the game where the cheating sender $\widehat{\mathsf{S}}$ is run against an honest receiver R with selection strategy Σ. Obviously, $\Pr[\,\mathbf{Game\text{-}0}\,] \;=\; \Pr\Big[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}\,\Big]$.

[5] This is a standard assumption we borrow from the literature on Identity-Based Encryption. The target group is usually a subgroup of a larger prime field. Thus, depending on implementation, it may be necessary to "hash" the data messages into this subgroup. Alternatively, one can extract a random pad from the element in the target group and use $\oplus$ to encrypt the message.

Game-1 : In this game the extractor E_1 for the first proof of knowledge is used to extract from $\widehat{\mathsf{S}}$ the element h such that $e(g,h) = H$. If the extractor fails, then the output of **Game-1** is $\perp$; otherwise, the execution of $\widehat{\mathsf{S}}$ continues as in the previous game, interacting with $\mathsf{R}(\Sigma)$. The difference between the two output distributions is given by the knowledge error of the PoK, i.e.,

$$\Pr[\,\textbf{Game-1}\,] - \Pr[\,\textbf{Game-0}\,] \;\leq\; \frac{1}{p}\;.$$

Game-2 : This game is identical to the previous one, except that during the transfer phase the value V sent by the receiver is replaced by picking a random v' and sending $V' \leftarrow A_1^v$. The witness $(v', 1)$ is used during the second PoK. Since V and V' are both uniformly distributed over $\mathbb{G}_1$, and by the perfect witness-indistinguishability of the PoK (implied by the perfect zero-knowledge property), we have that $\Pr[\,\textbf{Game-2}\,] \;=\; \Pr[\,\textbf{Game-1}\,]$.

Game-3 : In this game, we introduce an ideal-world sender $\widehat{\mathsf{S}}'$ which incorporates the steps from the previous game. Algorithm $\widehat{\mathsf{S}}'$ uses E_1 to extract h from $\widehat{\mathsf{S}}$, decrypts M_i^* as $B_i/e(h,A_i)$ for $i = 1,\ldots,N$ and submits $M_1^*,\ldots,M_N^*$ to the trusted party T. As in **Game-2**, during the transfer phase, $\widehat{\mathsf{S}}'$ feeds $V' \stackrel{\$}{\leftarrow} A_1^{v'}$ to $\widehat{\mathsf{S}}$ and uses $(v', 1)$ as a witness in the PoK. It plays the role of the verifier in the final PoM of W. If $\widehat{\mathsf{S}}$ convinces $\widehat{\mathsf{S}}'$ that W is correctly formed, then $\widehat{\mathsf{S}}'$ sends 1 to the trusted party, otherwise it sends 0. When $\widehat{\mathsf{S}}$ outputs its final state S_k, $\widehat{\mathsf{S}}'$ outputs S_k as well.

One can syntactically see that

$$\Pr[\,\textbf{Game-3}\,] \;=\; \Pr[\,\textbf{Game-2}\,] \;=\; \Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \stackrel{\$}{\leftarrow} \textbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}\,\right]\;.$$

Summing up, we have that the advantage of the distinguisher D is given by

$$\Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \stackrel{\$}{\leftarrow} \textbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}\,\right] - \Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \stackrel{\$}{\leftarrow} \textbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}\,\right] \;\leq\; \frac{1}{p}\;.$$

Sender security. The following theorem states the sender-security of our second construction.

Theorem 6. *If the $(N+1)$-SDH assumption and the $(N+1)$-PDDH assumptions associated to Pg hold, then the $\mathcal{OT}^N_{k\times 1}$ protocol depicted in Fig. 2 is sender-secure.*

Proof. Given a real cheating receiver $\widehat{\mathsf{R}}$, we construct an ideal-world cheating receiver $\widehat{\mathsf{R}}'$ such that no algorithm D can distinguish between the distributions $\textbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N,k,M_1,\ldots,M_N,\Sigma)$ and $\textbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N,k,M_1,\ldots,M_N,\Sigma)$. We again do so by considering a sequence of hybrid distributions and investigate the differences between successive ones.

Game-0 : This is the distribution corresponding to $\widehat{\mathsf{R}}$ being run against the honest sender $\mathsf{S}(M_1,\ldots,M_N)$. Obviously, we have that $\Pr[\,\textbf{Game-0}\,] \;=\; \Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \stackrel{\$}{\leftarrow} \textbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}\,\right]$.

Game-1 : This game differs from the previous one in that at each transfer the extractor E_2 of the second PoK is used to extract from $\widehat{\mathsf{R}}$ the witness (σ_i, v). If the extraction fails, **Game-1** outputs $\perp$. Because the PoK is perfect zero-knowledge, the difference on the distribution with the previous game is statistical (i.e., independent of the distinguisher's running time) and given by k times the knowledge error, or $\Pr[\,\textbf{Game-1}\,] - \Pr[\,\textbf{Game-0}\,] \;\leq\; k/p$. Note that the time required to execute these k extractions is k times the time of doing a single extraction, because the transfer protocols can only run sequentially, rather than concurrently. One would have to resort to concurrent zero-knowledge protocols [DNS04] to remove this restriction.

Game-2 : This game is identical to the previous one, except that **Game-2** returns $\perp$ if the extracted value $\sigma_i \notin \{1, \ldots, N\}$ during any of the transfers. One can see that in this case $s = V^{1/v}$ is a forged Boneh-Boyen signature on message σ_i. The difference between **Game-1** and **Game-2** is bounded by the following claim, which we prove below:

Claim (1). If the $(N+1)$-SDH assumption associated to Pg holds, then $\Pr[\,\textbf{Game-2}\,] - \Pr[\,\textbf{Game-1}\,]$ is negligible.

Game-3 : In this game the PoK of h in the initialization phase is replaced with a simulated proof using Sim_1, the value W returned in each transfer phase is computed as $W \leftarrow (B_{\sigma_i}/M_{\sigma_i})^v$, and the final PoM in the transfer phase is replaced by a simulated proof using Sim_3. Note that now the simulation of the transfer phase no longer requires knowledge of h. However, all of the simulated proofs are proofs of true statements and the change in the computation of W is purely conceptional. Thus by the perfect zero-knowledge property, we have that $\Pr[\,\textbf{Game-3}\,] \;=\; \Pr[\,\textbf{Game-2}\,]$.

Game-4 : Now the values $B_1, \ldots, B_N$ sent to $\widehat{\mathsf{R}}$ in the initialization phase are replaced with random elements from $\mathbb{G}_{\mathrm{T}}$. Now at this point, the second proof in the previous game is a simulated proof of a false statement. Intuitively, if these changes enable a distinguisher D to separate the experiments, then one can solve an instance of the SBDHI problem. This is caputed in the following claim:

Claim (2). If the $(N+1)$-PDDH assumption associated to Pg holds, then $\Pr[\,\textbf{Game-4}\,] - \Pr[\,\textbf{Game-3}\,]$ is negligible.

The ideal-world receiver $\widehat{\mathsf{R}}'$ can be defined as follows. It performs all of the changes to the experiments described in **Game-4** except that at the time of transfer, after having extracted the value of σ_i from $\widehat{\mathsf{R}}$, it queries the trusted party T on index σ_i to obtain message M_{σ_i}. It then uses this message to compute W. Syntactically, we have that

$$\Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}\,\right] \;=\; \Pr[\,\textbf{Game-4}\,]\;.$$

Summing up the above equations and inequalities yields that

$$\Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}\,\right] - \Pr\left[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}\,\right]$$

is negligible. The running time of $\widehat{\mathsf{R}}'$ is that of $\widehat{\mathsf{R}}$ plus that of $O(N^2)$ exponentiations, k extractions and k proof simulations, so is polynomial in κ.

It remains to prove the claims used in the proof above.

Proof (Claim (1)). We prove the claim by constructing an adversary A that breaks the unforgeability under weak chosen-message attack of the modified Boneh-Boyen signature scheme . By the security proof of [BB04], this directly gives rise to an expected polynomial-time adversary with non-negligible advantage in solving the $(N+1)$-SDH problem.

Given a cheating receiver $\widehat{\mathsf{R}}$ for that distinguishes between **Game-1** and **Game-2** with advantage ϵ_{St}, consider the forger A that outputs messages $M_1 = 1, \ldots, M_N = N$, and on input a public key y and signatures $A_1, \ldots, A_N$ runs the honest sender algorithm using these values for h and $A_1, \ldots, A_N$. At each transfer it uses E_2 to extract from $\widehat{\mathsf{R}}$ values (σ_i, v) such that $e(V, y) = e(V, g)^{-\sigma_i} e(g, g)^v$. (This extraction is guaranteed to succeed since we already eliminated failed extractions in the transition from **Game-0** to **Game-1**.) When $\sigma_i \notin \{1, \ldots, N\}$ then A outputs $s \leftarrow V^{1/v}$ as its forgery on message $M = \sigma_i$. The forger A wins whenever it extracts a value $\sigma_i \notin \{1, \ldots, N\}$ from $\widehat{\mathsf{S}}$. Its running time is that of $\widehat{\mathsf{R}}$ plus k times the running time of a single extraction, so polynomial in κ.

Proof (Claim (2)). Given an algorithm D with non-negligible probability in distinguishing **Game-2** and **Game-3**, consider the following algorithm A for the PDDH problem for $\ell = N + 1$. On input $(u, u^x, \ldots, u^{x^{N+1}}, V)$ and a vector $(T_1, \ldots, T_{N+1})$, A proceeds as follows. For ease of notation, let $T_0 = V$. Let f be the polynomial defined as $f(X) = \prod_{i=1}^{N}(X + i) = \sum_{i=0}^{N} c_i X^i$. Then A sets $g \leftarrow u^{f(x)} = \prod_{i=0}^{N}(u^{x^i})^{c_i}$ and $y \leftarrow g^x = \prod_{i=0}^{N}(u^{x^{i+1}})^{c_i}$. If f_i is the polynomial defined by $f_i(X) = f(X)/(X+i) = \sum_{j=0}^{N-1} c_{i,j} X^j$, then A can also compute the values $A_i = g^{1/(x+i)}$ as $A_i \leftarrow \prod_{j=0}^{N-1}(u^{x^j})^{c_{i,j}}$. It then sets $H \leftarrow V^{f(x)} = \prod_{i=0}^{N} T_i^{c_i}$, and computes $B_i = H^{1/(x+i)}$ as $B_i \leftarrow \prod_{j=0}^{N-1} T_i^{c_{i,j}}$, and continues the simulation of $\widehat{\mathsf{R}}$'s environment as in **Game-3** and **Game-4**, i.e., at each transfer extracting (σ_i, v), computing $W \leftarrow (B_{\sigma_i}/M_{\sigma_i})$ and simulating the PoM. When $\widehat{\mathsf{R}}$ outputs its final state R_k, algorithm A runs $b \xleftarrow{\$} \mathsf{D}(\varepsilon, R_k)$ and outputs b.

In the case that $T_i = V^{x^i}$ one can see that the environment that A created for $\widehat{\mathsf{S}}$ is exactly that of **Game-3**. In the case that $T_1, \ldots, T_N$ are random elements of $\mathbb{G}_{\mathrm{T}}$, then one can easily see that this environment is exactly that of **Game-4**. Therefore, if D has non-negligible advantage in distinguishing the outputs of **Game-3** and **Game-4**, then A has non-negligible advantage in solving the $(N+1)$-PDDH problem. The running time of A is at most that of the distinguisher D plus that of $O(N^2)$ exponentiations, of $k+1$ simulated proofs, and of k extractions.

Acknowledgements

The authors would like to thank Xavier Boyen, Christian Cachin, Anna Lysyanskaya, Benny Pinkas, Alon Rosen and the anonymous referees for their useful comments and discussions. Gregory Neven is a Postdoctoral Fellow of the Research Foundation Flanders (FWO-Vlaanderen). This work was supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT and Contract IST-2002-507591 PRIME.

References

[AIR01] W. Aiello, Y. Ishai, and O. Reingold. : Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT 2001*, p. 119–135.

[ANN06] M. Abdalla, C. Namprempre, and G. Neven. On the (im)possibility of blind message authentication codes. In *CT-RSA 2006*, p. 262–279.

[BB04] D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT 2004*, p. 56–73.

[BCR87] G. Brassard, C. Crépeau, and J.M. Robert. All-or-nothing disclosure of secrets. In *CRYPTO'86*, p. 234–238.

[BG92] M. Bellare and O. Goldreich. On defining proofs of knowledge. In *CRYPTO'92*, p. 390–420.

[BM90] M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. In *CRYPTO'89*, p. 547–557.

[BNPS03] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215, 2003.

[Bol03] A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In *PKC 2003*, p. 31–46.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, p. 62–73.

[Can00] R. Canetti. Security and composition of multi-party cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[CDM00] R. Cramer, I. Damgård, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *PKC 2000*, p. 354–372.

[Cha88] D. Chaum. Blind signature systems. U.S. Patent #4,759,063, 1988.

[CK90] C. Crépeau and J. Kilian. Weakening security assumptions and oblivious transfer. In *CRYPTO'88*, p. 2–7.

[CL04] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, p. 56–72.

[CNS07] J. Camenisch, G. Neven, and a. shelat. Simulatable Adaptive Oblivious Transfer. Cryptology ePrint Archive, 2007.

[CPP07] D. Catalano, D. Pointcheval, and T. Pornin. Trapdoor hard-to-invert group isomorphisms and their application to password-based authentication. To appear in *J. Cryptology*, 2007.

[CS97] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *CRYPTO'97*, p. 410–424.

[CT05] C.-K. Chu and W.-G. Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *PKC 2005*, p. 172–183.

[DNS04] C. Dwork, M. Naor, and A. Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.

[DY05] Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In *PKC 2005*, p. 416–431.

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[ElG85] T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[Gol04] O. Goldreich. Foundations of Cryptography, Volume 2. Cambridge University Press, 2004.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *19th ACM STOC*, p. 218–229.

[GO92] S. Goldwasser and R. Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent. In *CRYPTO'92*, p. 228–245.

[JLO97] A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (Extended abstract). In *CRYPTO'97*, p. 150–164.

[Kal05] Y. Kalai. Smooth projective hashing and two-message oblivious transfer. In *EUROCRYPT 2005*, p. 78–95.

[LL06] S. Laur and H. Lipmaa. On security of sublinear oblivious transfer. Cryptology ePrint Archive, 2006.

[MS03] D. Malkhi and Y. Sella. Oblivious transfer based on blind signatures. Technical Report 2003-31, Leibniz Center, Hebrew University, 2003.

[MSK02] S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. *IEICE Transactions Fundamentals*, E85-A(2):481–84, 2002.

[NP99a] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *31st ACM STOC*, p. 245–254, 1999.

[NP99b] M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. In *CRYPTO'99*, p. 573–590.

[NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *12th SODA*, p. 448–457, 2001.

[NP05] M. Naor and B. Pinkas. Computationally secure oblivious transfer. *J. Cryptology*, 18, 2005.

[OK04] W. Ogata and K. Kurosawa. Oblivious keyword search. *J. Complexity*, 20(2-3):356–371, 2004.

[Oka06] T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC 2006*, LNCS, p. 80–99.

[OS04] W. Ogata and R. Sasahara. k out of n oblivious transfer without random oracles. *IEICE Transactions*, 87-A(1):147–151, 2004.

[PS96] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT'96*, p. 387–398.

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[Wie83] S. Wiesner. Conjugate Coding. *SIGACT News*, 15, 1983. . 78–88.