An extended abstract of this paper appears in Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of Lecture Notes in Computer Science, pages 573–590, Springer-Verlag, 2007. This is the full version.

Simulatable Adaptive Oblivious Transfer

Jan Camenisch[1] Gregory Neven[2,3] abhi shelat[4]

[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
[4] University of Virginia, Charlottesville, Virginia

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 up to k times allowing the receiver to retrieve up to k messages of its choice 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).[1] 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 an adaptive k-out-ot-of-n OT. Notice that a cheating sender can always make the current transfer fail by sending bogus messages. However, we would not expect him to be able to cause failure based on some property of the receiver's selection. Of course, the sender 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

[1] 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 $O(\text{polylog}(N))$.

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 $\{1, \ldots, N\}$.

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 was proposed by 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 $\mathcal{OT}^N_{k\times 1}$ schemes satisfying half-simulation which require $O(k)$ computation and communication during the transfer stage. Our first generic $\mathcal{OT}^N_{k\times 1}$ 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 $\mathcal{OT}^N_1$ protocol (also in the random oracle model) based on Chaum's blind signatures [Cha88]. Their scheme could be seen as a $\mathcal{OT}^N_{k\times 1}$ 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.

$\mathcal{OT}^N_{k\times 1}$ can always be achieved by publishing commitments to the N data items, and executing k $\mathcal{OT}^N_1$ 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 in their input. 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] which allows the signer to choose the public key maliciously. In this work, we further strengthen the definition to *selective-failure blindness*. Intuitively, it prevents a cheating signer from making the user algorithm fail depending on the message that is being signed. This property seems important in practice, yet is not implied by any of the existing definitions. For example, consider a voting protocol where an administrator issues blind signatures on the voters' votes [FOO93]. If the scheme is not selective-failure blind, the administrator could for example let the protocol fail for votes for John Kerry, but let it proceed normally for votes for George W. Bush. Affected Kerry voters could complain, but by doing so they give up the privacy of their vote.

Selective-failure blindness 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 $b \xleftarrow{\$} \{0,1\}$. 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 $(\perp, \perp)$. (It is here that our definition is stronger than the existing ones: in the existing definition, the adversary is simply given $\perp$ if either algorithm fails.) 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 $\mathcal{OT}^N_{k\times 1}$ 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 $\mathsf{R_T}(R_{i-1}, \sigma_i)$ 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 $\mathcal{OT}^N_{k\times 1}$ 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 $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$ proceeds as follows. $\widehat{\mathsf{S}}$ is given messages $(M_1, \ldots, M_N)$ as input and interacts with $\widehat{\mathsf{R}}$ without input. 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 $(R_i, M'_{\sigma_i}) \xleftarrow{\$} \widehat{\mathsf{R}}(R_{i-1}, \sigma_i)$, where $\sigma_i \in \{1, \ldots, N\}$ is a message index. Both algorithms update their states to S_i and R_i, respectively. Note that M'_{σ_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 algorithm that 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, and returns the list of received messages $R_k = (M'_{\sigma_1}, \ldots, M'_{\sigma_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$, 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 $\widehat{\mathsf{R}}'(\sigma_i)$. 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 the algorithm that sends messages $M_1, \ldots, M_N$ to the trusted party in the initialization phase, sends $b_i = 1$ in all transfer phases, and outputs $S_k = \varepsilon$ as its final state. Define the honest ideal receiver R' as the algorithm that at each transfer submits the real selection index σ_i to the trusted party, and that outputs the list of all received messages $R_k = (M'_{\sigma_1}, \ldots, M'_{\sigma_k})$ as its final state.

Note that the sender's bit b_i models its ability to make the current transfer fail. However, the sender's decision to do so is independent of the index σ'_i that is being queried by the receiver. This captures the strongest notion of "coherence" as envisaged by [LL06], and excludes schemes like [NP99b] that allow the sender to cause selective failure.

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 $N \in \{1, \ldots, N_m(\kappa)\}$, any $k \in \{1, \ldots, N\}$, any messages $M_1, \ldots, M_N$, and any indices $\sigma_1, \ldots, \sigma_k \in \{1, \ldots, N\}$, the advantage of any PPT distinguisher in distinguishing the distributions

$$\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k) \quad \text{and} \quad \mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$$

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 $N \in \{1, \ldots, N_m(\kappa)\}$, any $k \in \{1, \ldots, N\}$, any messages $M_1, \ldots, M_N$, and any indices $\sigma_1, \ldots, \sigma_k \in \{1, \ldots, N\}$, the advantage of any PPT distinguisher in distinguishing the distributions

$$\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k) \quad \text{and} \quad \mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$$

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 Figure 1. The security of the oblivious transfer scheme follows from that of the blind signature scheme. In particular, Theorem 3.1 states that sender security is implied by the one-more unforgeability of $\mathcal{BS}$, while Theorem 3.2 states that receiver security follows from the selective-failure blindness of $\mathcal{BS}$. Note that correctness follows from the uniqueness of $\mathcal{BS}$.

Theorem 3.1 If the blind signature scheme $\mathcal{BS}$ is unforgeable, then the $\mathcal{OT}^N_{k\times 1}$ depicted in Figure 1 is sender-secure in the random oracle model.

Proof: For any real-world cheating receiver $\widehat{\mathsf{R}}$, consider the ideal-world receiver $\widehat{\mathsf{R}}'$ that works as follows. $\widehat{\mathsf{R}}'$ generates a fresh key pair $(pk, sk) \xleftarrow{\$} \mathsf{Kg}$ for the blind signature scheme $\mathcal{BS}$ and chooses

INITIALIZATION

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

TRANSFER

$\underline{\mathsf{S_T}(S_{i-1}):}$		$\underline{\mathsf{R_T}(R_{i-1}, \sigma_i):}$
Parse S_{i-1} as sk		Parse R_{i-1} as $(pk, C_1, \ldots, C_N)$
Run protocol $\mathsf{Sign}(sk)$	$\longleftrightarrow$	Run protocol $s_i \xleftarrow{\$} \mathsf{User}(pk, \sigma_i)$
		If $\mathsf{Vf}(pk, \sigma_i, s_i) = 0$ then $M_{\sigma_i} \leftarrow \perp$
		Else $M'_{\sigma_i} \leftarrow C_{\sigma_i} \oplus H(\sigma_i, s_i)$
Output $S_i = S_{i-1}$		Output $(R_i = R_{i-1}, M'_{\sigma_i})$

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

random strings $C_1, \ldots, C_N \xleftarrow{\$} \{0,1\}^\ell$. It then feeds the string $(pk, C_1, \ldots, C_N)$ as input to $\widehat{\mathsf{R}}$ to obtain initial state R_0.

During the transfer phase, when $\widehat{\mathsf{R}}$ engages in a transfer protocol, $\widehat{\mathsf{R}}'$ simulates the honest sender by executing the blind signature protocol as prescribed by $\mathsf{Sign}(sk)$. To answer random oracle queries, $\widehat{\mathsf{R}}'$ maintains an initially empty associative array $\mathrm{HT}[\cdot]$ and a counter ctr. When $\widehat{\mathsf{R}}$ performs a random oracle query $\mathrm{H}(x)$, $\widehat{\mathsf{R}}'$ responds with $\mathrm{HT}[x]$, or proceeds as follows if this entry is undefined:

If $x = (i, s)$ and $\mathsf{Vf}(pk, i, s) = 1$ and $i \in [1, N]$ then
$\quad ctr \leftarrow ctr + 1$; If $ctr > k$ then abort
$\quad$ Obtain M_i from the ideal functionality
$\quad \mathrm{HT}[x] \leftarrow M_i \oplus C_i$
else $\mathrm{HT}[x] \xleftarrow{\$} \{0,1\}^\ell$.

When eventually $\widehat{\mathsf{R}}$ outputs its final state R_k, $\widehat{\mathsf{R}}'$ halts with the same output R_k. The running time t' of $\widehat{\mathsf{R}}'$ is that of $\widehat{\mathsf{R}}$ plus the time of a key generation, k signing interactions and up to q_{H} signature verifications. It is clear that if $\widehat{\mathsf{R}}'$ does not abort, then $\widehat{\mathsf{R}}'$ provides $\widehat{\mathsf{R}}$ with a perfect simulation of the $\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_k, \sigma_1, \ldots, \sigma_k)$ experiment, so no distinguisher D has advantage greater than zero in distinguishing $\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}$ from $\mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}$.

We now show that if there exists an algorithm $\widehat{\mathsf{R}}$ that causes $\widehat{\mathsf{R}}'$ to abort with non-negligible probability, then there exists a forger F with non-negligible advantage in breaking $\mathcal{BS}$. Algorithm F simulates the environment of $\widehat{\mathsf{R}}$ in a similar way as $\widehat{\mathsf{R}}'$, except that (1) it relays messages between its signing oracle and $\widehat{\mathsf{R}}$ to simulate transfer queries, and (2) rather than aborting when $ctr > k$, it outputs all $k+1$ valid message-signature pairs (i, s) that $\widehat{\mathsf{R}}$ submitted to the random oracle. Since $\widehat{\mathsf{R}}$ can engage in at most k transfer protocols, $\widehat{\mathsf{R}}'$ outputs $k+1$ valid signatures after at most k signature queries, and hence wins the one-more unforgeability game. ■

Theorem 3.2 If the blind signature scheme $\mathcal{BS}$ is selective-failure blind, then the $\mathcal{OT}^N_{k\times 1}$ scheme depicted in Figure 1 is receiver-secure in the random oracle model.

Proof: We have to show that for any real-world cheating sender $\widehat{\mathsf{S}}$, there exists an ideal-world sender $\widehat{\mathsf{S}}'$ whose output is indistinguishable from that of $\widehat{\mathsf{S}}$. Consider the ideal-world sender $\widehat{\mathsf{S}}'$ that on input $(M_1, \ldots, M_N)$ runs $\widehat{\mathsf{S}}(M_1, \ldots, M_N)$, simulating its random oracle queries by returning random values. Let $(pk, C_1, \ldots, C_N)$ be the outgoing message produced by $\widehat{\mathsf{S}}$ during the initialization phase. For all random oracle queries $H(i, s)$ with $1 \leq i \leq N$ made by $\widehat{\mathsf{S}}$, $\widehat{\mathsf{S}}'$ checks whether $\mathsf{Vf}(pk, i, s) = 1$. If so, then it sets $M'_i \leftarrow C_i \oplus H(i, s)$. For all $1 \leq j \leq N$ such that M'_j has not been defined by this procedure, it assigns a random value $M'_j \xleftarrow{\$} \{0,1\}^\ell$. Algorithm $\widehat{\mathsf{S}}'$ submits $(M'_1, \ldots, M'_N)$ to the trusted party.

To subsequently handle the k transfers, $\widehat{\mathsf{S}}'$ sets $R_0 \leftarrow (pk, C_1, \ldots, C_N)$ and at the i-th transfer simulates the environment of $\widehat{\mathsf{S}}$ by running $R_i \leftarrow \mathsf{R_T}(R_{i-1}, 1)$, i.e., by always running the honest receiver that queries for the message with index one. Remember that $\widehat{\mathsf{S}}'$ is not given the selection indices $(\sigma_1, \ldots, \sigma_k)$ as input, so it cannot run $\mathsf{R_T}$ on the real index σ_i. If the output of $\mathsf{R_T}$ is $\perp$, then $\widehat{\mathsf{S}}'$ sends $b_i = 0$ to the trusted party, indicating that this query should be aborted; otherwise, it sends $b_i = 1$. Random oracle queries of the form $H(i, s)$ with $1 \leq i \leq N$ and $\mathsf{Vf}(pk, i, s) = 1$ in this phase are answered with $C_i \oplus M'_i$; all other random oracle queries are answered with random values. At the end of the k-th query, $\widehat{\mathsf{S}}$ outputs its final state S_k; the ideal sender $\widehat{\mathsf{S}}'$ outputs the same string S_k.

We use a hybrid proof to analyze the advantage of an algorithm D in distinguishing between $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}$ and $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}$. For $0 \leq i \leq k$, let $\widehat{\mathsf{S}}'_i$ be an algorithm that simulates the environment of $\widehat{\mathsf{S}}$ in a similar way as $\widehat{\mathsf{S}}'$, but that uses $R_i \xleftarrow{\$} \mathsf{R_T}(R_{i-1}, 1)$ for the first i transfers, and that uses $R_i \xleftarrow{\$} \mathsf{R_T}(R_{i-1}, \sigma_i)$ for the remaining $k-i$ transfers. Let the output of experiment **Game-i** contain the final states of $\widehat{\mathsf{S}}'_i$ and of the honest ideal-world receiver $\mathsf{R}'(\sigma_1, \ldots, \sigma_k)$ after interacting with each other through a trusted party T as in the ideal experiment. It is easy to see that $\mathbf{Game\text{-}0} = \mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}$ and that $\mathbf{Game\text{-}k} = \mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}$. If there exists an algorithm D that distinguishes between $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}$ and $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}$ with non-negligible advantage ϵ, then there must exist an index $0 \leq i \leq k$ such that D distinguishes between **Game-i** and $\mathbf{Game\text{-}(i+1)}$ with probability at least ϵ/k.

Given this distinguisher D, we show how to construct an adversary A against the selective-failure blindness of $\mathcal{BS}$. Algorithm A runs $\widehat{\mathsf{S}}$, answering random oracle queries and extracting messages $M_1, \ldots, M_N$ in the same way as described for $\widehat{\mathsf{S}}'$. It simulates the j-th transfer for $1 \leq j \leq i$ using $\mathsf{R_T}(\cdot, 1)$, setting $M'_j = M_{\sigma_j}$ if the transfer succeeds, and setting $M'_j = \perp$ if it doesn't. For the $(i+1)$-st transfer A uses its first user oracle to simulate the receiver. More particularly, it outputs $pk, m_0 = \sigma_i, m_1 = 1$ as the public key and messages on which it wishes to be challenged, and relays messages between $\widehat{\mathsf{S}}$ and its first oracle that implements $\mathsf{User}(pk, m_b)$ for some hidden value $b \in \{0,1\}$. With the second oracle, that implements $\mathsf{User}(pk, m_{1-b})$, it interacts in an arbitrary way (most likely causing it to fail) until it receives signatures (s_0, s_1). If $s_0 = \perp$ then it sets $M'_{i+1} = \perp$, otherwise it sets $M'_{i+1} = M_{\sigma_{i+1}}$. For the remaining transfers $i+2 \leq j \leq k$, it uses $\mathsf{R_T}(\cdot, \sigma_j)$ to simulate the receiver, setting $M'_j = M_{\sigma_j}$ if the transfer succeeds and $M'_j = \perp$ if it doesn't. When $\widehat{\mathsf{S}}$ outputs its final state S_k, A feeds the tuple $(S_k, (M'_1, \ldots, M'_k))$ to D. One can check that if $b = 0$ then this tuple is distributed according to **Game-i**, while if $b = 1$ then it is distributed according to $\mathbf{Game\text{-}(i+1)}$. Algorithm A therefore can therefore use D's output to win the blindness game with advantage at least ϵ/k. ■

3.2 Instantiations

Of all the existing blind signature schemes in the literature, we were only able to discover two that are unique, namely the schemes by Chaum [Cha88, BNPS03] and Boldyreva [Bol03]. Both are efficient two-round schemes, yielding 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 $\gcd(e, n) = 1$ [CPP06].[2] The authors of [OK04] overlooked this need, leading to easy attacks on the receiver 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 and that the generator is of full order.

3.3 Oblivious Keyword Search

Oblivious keyword search [CGN98, OK04] is a generalization of oblivious transfer where messages are indexed by keywords rather than by consecutive numbers. The sender owns a database of message-keyword pairs $(M_1, w_1), \ldots, (M_N, w_N)$. (We can assume without loss of generality that every keyword appears at most once.) At each transfer, the receiver chooses a keyword w, and receives M_i if $w = w_i$ for some $1 \le i \le N$, or receives $\perp$ otherwise. The receiver does not learn any other information about the database, and the sender does not learn anything about the receiver's selection w.

The oblivious transfer protocol of Figure 1 can be easily modified into an oblivious keyword search protocol by signing keywords instead of index numbers, i.e., by taking $s_i \leftarrow \mathsf{Sign}(sk, w_i)$. Messages are encrypted as $C_i \leftarrow H(w_i, s_i) \oplus 0^{2\kappa} \| M_i$. In the transfer phase, the receiver obtains a blind signature s on keyword w, and for all $1 \le i \le N$ computes $C_i \oplus H(w, s)$ until the first 2κ bits of the obtained bitstring are all zeroes. Instantiating our generic construction with Chaum's blind signatures yields the oblivious keyword search scheme due to Ogata-Kurosawa [OK04], instantiating with Boldyreva's blind signature scheme yields a new oblivious keyword search scheme based on the one-more CDH assumption.

The reason for padding messages with 2κ zeroes instead of κ is the following. Suppose messages are padded with ℓ zeroes to detect correct decryption. If a cheating sender can come up with w, s, w', s' so that the first ℓ bits of $H(w, s)$ and $H(w', s')$ are the same, then the same ciphertext C will decrypt correctly under both w and w'. In the proof of receiver security, the ideal-world sender can then no longer extract a unique decryption of C, because there are now two different possibilities. We therefore need finding collisions on the first ℓ bits of the hash output to be hard, so by the birthday paradox we need $\ell = 2\kappa$ to obtain $O(2^\kappa)$ security.

4 Simulatable Adaptive OT in the Standard Model

4.1 Preliminaries

Computational assumptions. Our protocol presented in this section requires bilinear groups and

[2] Anna Lysyanskaya suggested to let 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.

associated hardness assumptions. Let Pg be a pairing group generator that on input 1^κ outputs descriptions of multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$ 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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, 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 4.1 [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 $(\mathbb{G}_1, \mathbb{G}_T) \xleftarrow{\$} \mathsf{Pg}(1^\kappa)$, $g \xleftarrow{\$} \mathbb{G}_1^*$ and $x \xleftarrow{\$} \mathbb{Z}_p$, outputs a pair $(c, g^{1/(x+c)})$ where $c \in \mathbb{Z}_p$ is negligible in κ.

Definition 4.2 [Power Decisional Diffie-Hellman Assumption] We say that the ℓ-PDDH 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 $(\mathbb{G}_1, \mathbb{G}_T) \xleftarrow{\$} \mathsf{Pg}(1^\kappa)$, $g \xleftarrow{\$} \mathbb{G}_1^*$, $x \xleftarrow{\$} \mathbb{Z}_p$, $H \xleftarrow{\$} \mathbb{G}_T$, distinguishes the vector $T = (H^x, H^{x^2}, \ldots, H^{x^\ell})$ from a random vector $T \xleftarrow{\$} \mathbb{G}_T^\ell$ is negligible in κ.

Evidence of the hardness of this new problem is presented in Appendix C.

Boneh-Boyen signatures. We use the following slight modification of the weakly-secure signature scheme by Boneh and Boyen [BB04]. 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 $(g, y = g^x)$ where g is a random generator of $\mathbb{G}_1$. 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 outputting 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. The proof is provided in Appendix A for completeness.

Zero-knowledge proofs and Σ-protocols. 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 output of $\widehat{\mathsf{V}}(\alpha)$ after interacting with $\mathsf{P}(\beta)$ and the output of $\mathsf{Sim}^{\widehat{\mathsf{V}}(\alpha)}(\alpha)$ are identically distributed.

A Σ-protocol is a proof system (P, V) where the conversation is of the form (a, c, z), where a and z are are computed by P, and c is a challenge chosen at random by V. The verifier accepts if $\phi(\alpha, a, c, z) = 1$ for some efficiently computable predicate ϕ. Given two accepting conversations (a, c, z) and (a, c', z') for $c \neq c'$, one can efficiently compute a witness β. Moreover, there exists a polynomial-time simulator Sim that on input α and a random string c outputs an accepting conversation (a, c, z) for α that is perfectly indistinguishable from a real conversation between $\mathsf{P}(\beta)$ and $\mathsf{V}(\alpha)$.

For a relation $R = \{(\alpha, \beta)\}$ with Σ-protocol (P, V), the *commitment relation* $R' = \{(\alpha, a), (c, z)\}$ holds if $\phi(\alpha, a, c, z) = 1$. If both R and R' have Σ-protocols, then Cramer et al. [CDM00] show how

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\textsc{Initialization}} \\
\underline{\mathsf{S_I}(1^\kappa, M_1, \ldots, M_N) :} & & \underline{\mathsf{R_I}(1^\kappa) :} \\
(\mathbb{G}_1, \mathbb{G}_\mathrm{T}) \xleftarrow{\$} \mathsf{Pg}(1^\kappa) & & \\
g, h \xleftarrow{\$} \mathbb{G}_1^* \; ; \; H \leftarrow e(g,h) & & \\
x \xleftarrow{\$} \mathbb{Z}_p \; ; \; y \leftarrow g^x \; ; \; pk \leftarrow (g, H, y) & & \\
\text{For } i = 1, \ldots, N \text{ do} & & \\
\quad A_i \leftarrow g^{1/(x+i)} & & \\
\quad B_i \leftarrow e(h, A_i) \cdot M_i & & \\
\quad C_i \leftarrow (A_i, B_i) & \xrightarrow{\quad pk, C_1, \ldots, C_N \quad} & \\
S_0 \leftarrow (h, pk) & \xrightarrow{\quad PoK\{(h) :\ H = e(g,h)\} \quad} & R_0 \leftarrow (pk, C_1, \ldots, C_N) \\
\multicolumn{3}{c}{\textsc{Transfer}} \\
\underline{\mathsf{S_T}(S_{i-1}) :} & & \underline{\mathsf{R_T}(R_{i-1}, \sigma_i) :} \\
 & & v \xleftarrow{\$} \mathbb{Z}_p \; ; \; V \leftarrow (A_{\sigma_i})^v \\
 & \xleftarrow{\quad V \quad} & \\
 & \xleftarrow{\quad PoK\{(\sigma_i, v) : e(V,y) = e(V,g)^{-\sigma_i} e(g,g)^v\} \quad} & \\
W \leftarrow e(h, V) & \xrightarrow{\quad W \quad} & \\
 & \xrightarrow{\quad PoM\{(h) : H = e(g,h) \land W = e(h,V)\} \quad} & \\
S_i = S_{i-1} & & M \leftarrow B_{\sigma_i} / (W^{1/v}) \\
 & & R_i = R_{i-1}
\end{array}
$$

Figure 2: Our $\mathcal{OT}^N_{k\times 1}$ 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 Appendix B.

to construct a four-move perfect zero-knowledge PoK for R with knowledge error $\kappa = 1/|C|$, where C is the space from which the challenge c is drawn.

4.2 The Protocol

Our protocol in the standard model is depicted in Figure 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 Appendix B. We assume that the messages M_i are elements of the target group $\mathbb{G}_\mathrm{T}$.[3] 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 $\mathbb{G}_\mathrm{T}$ of M_i under public key H. But instead of using random elements from $\mathbb{G}_\mathrm{T}$ 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.

The relation of this protocol to blind signatures is not as explicit as in our random-oracle construction, but it could still be seen as being underlain by a somewhat "limited" blind signature

[3]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.

scheme. Namely, consider the scheme with public key $(g, H, y, A_1, \ldots, A_N)$ and corresponding secret key $\alpha = \log_g h = \log_{e(g,g)} H$, where the signature of a message $M \in \{1, \ldots, N\}$ is given by $s = (A_M)^\alpha$. The signing protocol would be a variation on the transfer phase of our OT scheme where the user is given $W = V^\alpha$ rather than $W = e(h, V)$. Verification is done by checking that $e(s, yg^M) = H$. The scheme has the obvious disadvantage that the public key is linear in the size of the message space; we therefore do not further study its properties here.

4.3 Security

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 4.3 The $\mathcal{OT}^N_{k\times 1}$ protocol in Figure 2 is statistically 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 $\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$ and $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \sigma_1, \ldots, \sigma_k)$. 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 that queries for index σ_i in the i-th transfer. Obviously,

$$\Pr[\,\mathbf{Game\text{-}0}\,] \;=\; \Pr\Big[\,\mathsf{D}(X) = 1 \;:\; X \xleftarrow{\$} \mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}\,\Big]\;.$$

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 $\mathsf{R}(\sigma_1), \ldots, \mathsf{R}(\sigma_k)$. The difference between the two output distributions is given by the knowledge error of the PoK, i.e.,

$$\Pr[\,\mathbf{Game\text{-}1}\,] - \Pr[\,\mathbf{Game\text{-}0}\,] \;\le\; \frac{1}{p}\;.$$

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 $i = 1, \dots, N$ and submits $M_1^*, \dots, M_N^*$ 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

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

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

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

■

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

Theorem 4.4 If the $(N+1)$-SDH assumption and the $(N+1)$-PDDH assumptions associated to Pg hold, then the $\mathcal{OT}_{k\times 1}^{N}$ protocol depicted in Figure 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 $\textbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \dots, M_N, \sigma_1, \dots, \sigma_k)$ and $\textbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N, k, M_1, \dots, M_N, \sigma_1, \dots, \sigma_k)$. 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 in interaction with the honest sender $\mathsf{S}(M_1, \dots, M_N)$. Obviously, we have that

$$\Pr[\,\textbf{Game-0}\,] \;=\; \Pr\Big[\,\mathsf{D}(X) = 1 \;:\; X \stackrel{\$}{\leftarrow} \textbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}\,\Big] \;.$$

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

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

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 4.5 If the $(N+1)$-SDH assumption associated to Pg holds, then

$$\Pr[\,\mathbf{Game\text{-}2}\,] - \Pr[\,\mathbf{Game\text{-}1}\,]$$

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

$$\Pr[\,\mathbf{Game\text{-}3}\,] \;=\; \Pr[\,\mathbf{Game\text{-}2}\,]\;.$$

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 PDDH problem. This is captured in the following claim:

Claim 4.6 If the $(N+1)$-PDDH assumption associated to Pg holds, then

$$\Pr[\,\mathbf{Game\text{-}4}\,] - \Pr[\,\mathbf{Game\text{-}3}\,]$$

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 $M_{\sigma_i'}$. It then uses this message to compute W. Syntactically, we have that

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

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 the security parameter. ∎

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

Proof of Claim 4.5: 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 given in Appendix A, 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**, 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 the security parameter. ■

Proof of Claim 4.6: 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 $b \stackrel{\$}{\leftarrow} \mathsf{D}(\varepsilon, R_k)$ 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, Markulf Kohlweiss, 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] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

[ANN06] Michel Abdalla, Chanathip Namprempre, and Gregory Neven. On the (im)possibility of blind message authentication codes. In *Topics in Cryptology – CT-RSA 2006*, Lecture Notes in Computer Science, pages 262–279, San Jose, CA, USA, February 13–17, 2006. Springer-Verlag, Berlin, Germany. (Cited on pages 5 and 10.)

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 11.)

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on pages 22 and 23.)

[BCR87] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 234–238, Santa Barbara, CA, USA, August 1987. Springer-Verlag, Berlin, Germany. (Cited on pages 1 and 4.)

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 16–20, 1992. Springer-Verlag, Berlin, Germany. (Cited on page 11.)

[BM90] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 547–557, Santa Barbara, CA, USA, August 20–24, 1990. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, 2003. (Cited on page 9.)

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46, Miami, USA, January 6–8, 2003. Springer-Verlag, Berlin, Germany. (Cited on pages 9 and 10.)

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–372, Melbourne, Victoria, Australia, January 18–20, 2000. Springer-Verlag, Berlin, Germany. (Cited on pages 11, 12 and 20.)

[CGN98] Benny Chor, Niv Gilboa, and Moni Naor. Private information retrieval by keywords. Cryptology ePrint Archive, Report 1998/003, 1998. `http://eprint.iacr.org/`. (Cited on page 10.)

[Cha88] David Chaum. Blind signature systems. U.S. Patent #4,759,063, July 1988. (Cited on pages 5 and 9.)

[CK90] Claude Crépeau and Joe Kilian. Weakening security assumptions and oblivious transfer. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 2–7, Santa Barbara, CA, USA, August 21–25, 1990. Springer-Verlag, Berlin, Germany. (Cited on page 1.)

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

[CPP06] Dario Catalano, David Pointcheval, and Thomas Pornin. Trapdoor hard-to-invert group isomorphisms and their application to password-based authentication. *Journal of Cryptology*, 2006. To appear, available from `http://www.di.ens.fr/~pointche/`. (Cited on page 10.)

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, CA, USA, August 17–21, 1997. Springer-Verlag, Berlin, Germany. (Cited on page 12.)

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In Serge Vaudenay, editor, *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 172–183, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on pages 5 and 10.)

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *Journal of the ACM*, 51(6):851–898, 2004. (Cited on page 14.)

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on pages 11 and 12.)

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the Association for Computing Machinery*, 28(6):637–647, 1985. (Cited on pages 1 and 4.)

[ElG85] Taher ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985. (Cited on page 12.)

[FOO93] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Josef Pieprzyk, editors, *Advances in Cryptology – AUSCRYPT ' 92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer-Verlag, Berlin, Germany, 1993. (Cited on page 5.)

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *19th Annual ACM Symposium on Theory of Computing*, pages 218–229, New York City, New York, USA, May 25–27, 1987. ACM Press. (Cited on page 1.)

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 228–245, Santa Barbara, CA, USA, August 16–20, 1992. Springer-Verlag, Berlin, Germany. (Cited on page 5.)

[Gol04] Oded Goldreich. *Foundations of Cryptography, Volume 2.* Cambridge University Press, 2004. (Cited on pages 3 and 4.)

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (Extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164, Santa Barbara, CA, USA, August 17–21, 1997. Springer-Verlag, Berlin, Germany. (Cited on page 5.)

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 78–95, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

[LL06] Sven Laur and Helger Lipmaa. On security of sublinear oblivious transfer. Cryptology ePrint Archive, 2006. `http://eprint.iacr.org/`. (Cited on pages 4 and 7.)

[MS03] Dahlia Malkhi and Yaron Sella. Oblivious transfer based on blind signatures. Technical Report 2003-31, Leibniz Center, Hebrew University, 2003. `http://leibniz.cs.huji.ac.il/tr/591.ps`. (Cited on page 5.)

[NP99a] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *31st Annual ACM Symposium on Theory of Computing*, pages 245–254, Atlanta, Georgia, USA, May 1–4, 1999. ACM Press. (Cited on page 4.)

[NP99b] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany. (Cited on pages 1, 2, 4 and 7.)

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM-SIAM. (Cited on page 4.)

[NP05] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18, 2005. (Cited on pages 2 and 5.)

[OK04] Wakaha Ogata and Kaoru Kurosawa. Oblivious keyword search. *Journal of Complexity* , 20(2-3):356–371, 2004. (Cited on pages 5 and 10.)

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, page ?, New York, NY, USA, March 4–7, 2006. Springer-Verlag, Berlin, Germany. (Cited on page 5.)

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, Saragossa, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany. (Cited on page 5.)

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981. (Cited on pages 1 and 4.)

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980. (Cited on page 24.)

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer-Verlag, Berlin, Germany. (Cited on page 22.)

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. (Cited on page 4.)

A Proof of Modified Boneh-Boyen Signatures

Given a forger F breaking the weak chosen-message security of the signature scheme described in Section 4.1, consider the following algorithm A solving the $(\ell+1)$-SDH problem. When given as input values $g, g^x, g^{x^2}, \ldots, g^{x^{\ell+1}}$, algorithm A runs F until it outputs messages $m_1, \ldots, m_\ell$. If $x = m_i$ for some $1 \leq i \leq \ell$ then A can trivially solve the $(\ell+1)$-SDH problem. Consider the polynomial $f(X) = \prod_{i=1}^{\ell}(X + m_i) = \sum_{i=0}^{\ell} \alpha_i X^i$. Algorithm A computes $g' \leftarrow g^{f(x)} = \prod_{i=0}^{\ell} \left(g^{x^i}\right)^{\alpha_i}$ and $y \leftarrow g^{xf(x)} = \prod_{i=0}^{\ell} \left(g^{x^{i+1}}\right)^{\alpha_i}$. It feeds (g', y) as the public key to F. To compute the signatures for $m_1, \ldots, m_\ell$, consider for each $1 \leq i \leq \ell$ the polynomial $f_i(X) = f(X)/(X + m_i) = \prod_{j \neq i}(X + m_j) = \sum_{j=0}^{\ell-1} \beta_j X^j$. Then A can compute the signature as $s_i \leftarrow g^{f_i(x)} = \prod_{j=0}^{\ell-1} \left(g^{x^j}\right)^{\beta_j}$.

Eventually, F outputs a message-signature pair (m, s) so that $m \notin \{m_1, \ldots, m_\ell\}$ and $e(s, y \cdot (g')^m) = e(g', g')$. If $m = -x$ then A can trivially solve the $(\ell+1)$-SDH problem. Otherwise, since $g' = g^{f(x)}$

and $y = g^{xf(x)}$ we have that

$$s = (g')^{1/(x+m)} = g^{f(x)/(x+m)} .$$

Let $\gamma(X) = \sum_{i=0}^{\ell-1} \gamma_i X^i$ be the polynomial such that $f(X) = (X + m) \cdot \gamma(X) + \gamma^*$ for some $\gamma^* \in \mathbb{Z}_p$. We have that $s = g^{\gamma(x)} \cdot g^{\gamma^*/(x+m)}$, so that when A computes

$$w \leftarrow \left(s \;/\; \prod_{i=0}^{\ell-1} (g^{x^i})^{\gamma_i} \right)^{1/\gamma^*}$$

we have that $w = g^{1/(x+m)}$. Algorithm A outputs (m, w) as its solution to the $(\ell + 1)$-SDH problem. The advantage of A in solving the $(\ell + 1)$-SDH problem is equal to that of F in breaking the weak security of the signature scheme.

B Proof of Knowledge Protocols

Cramer, Damgård, and MacKenzie [CDM00] present a framework for constructing four-round perfect zero-knowledge proofs of knowledge for a special class of languages that have efficient Σ-protocols—which in particular, includes the three discrete-log-based languages used in our protocol. Two remarkable properties of their construction are that it is unconditional, i.e., the protocol does not require any additional computational assumptions, and that the extraction error is exponentially small.

The CDM construction for a relationship $R = \{(\alpha, \beta)\}$ uses both a sigma protocol for R as well as a sigma protocol for the *commitment relationship* $R' = \{((\alpha, a), (c, z))\}$. Informally, this commitment relationship includes all pairs (α, a) for which there exists a witness (c, z) such that (a, c, z) is an accepting sigma-protocol transcript on the instance α. (It is called the commitment relationship since (α, a) can be viewed as a commitment to c when the committer does not know β.) Given a proof system (P, V) (and a corresponding simulator Sim) for R and a protocol $(\mathsf{P}', \mathsf{V}')$ (and a corresponding simulator Sim') for R', the CDM construction works as follows: in the first phase, the verifier commits to a challenge e by running the simulator $\mathsf{Sim}(\alpha, e)$ to generate a pair (a, e, z) and sending (α, a) to the prover. The verifier then runs the sigma protocol $\mathsf{P}'((\alpha, a), (e, z))$ with the prover who runs $\mathsf{V}'(\alpha, a)$ in order to prove knowledge of a witness of instance (α, a) for the R' relation. Then the prover uses the standard "OR-Σ-protocol" P_{OR} to prove that it either knows a witness β for α in R, or a witness (e, z) for (α, a) in R'. The special relationship between R and R' enables them to prove all of the incredible properties of this protocol. By merging rounds, this construction can be shortened to four rounds.

For the special case of knowledge of a discrete logarithm, Cramer, Damgård, and MacKenzie offer a specially optimized proof which we include below for completeness. We then give the Σ-protocols for R and R' for the two other languages for which we need proofs of knowledge.

B.1 Proof of Knowledge of a Pairing Preimage

Rather than giving the full zero-knowledge protocol, we give the Σ-protocols for the relation R and its commitment relation R'. The CDM construction can be applied to turn these components into a perfectly zero-knowledge proof system.

Σ-PROTOCOL FOR $\mathrm{PoK}\{(h) : H = e(g, h)\}$

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathrm{T}$, instance $H \in \mathbb{G}_\mathrm{T}$.

Prover's Input: $h \in \mathbb{G}_1$ s.t. $H = e(g, h)$.

$\mathsf{P} \xrightarrow{a} \mathsf{V}$: Prover picks $r \xleftarrow{\$} \mathbb{G}_1$ and sends $a = e(g, r)$.

$\mathsf{P} \xleftarrow{c} \mathsf{V}$: Verifier sends a random challenge $c \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z} \mathsf{V}$: Prover sends $z \leftarrow r \cdot h^{-c}$.

V: Verifier checks that $a \stackrel{?}{=} e(g, z) \cdot H^c$.

Σ-PROTOCOL FOR COMMITMENT RELATIONSHIP

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_{\mathrm{T}}$, instance $(H, a) \in \mathbb{G}_{\mathrm{T}}^2$.

Prover's Input: $(c, z) \in \mathbb{Z}_p \times \mathbb{G}_1$ such that $a = e(g, z) \cdot H^c$.

$\mathsf{P} \xrightarrow{a'} \mathsf{V}$: Prover picks $r_1 \xleftarrow{\$} \mathbb{Z}_p$, $r_2 \xleftarrow{\$} \mathbb{G}_1$ and sends $a' \leftarrow H^{r_1} \cdot e(g, r_2)$.

$\mathsf{P} \xleftarrow{c'} \mathsf{V}$: Verifier sends a random challenge $c' \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z_1', z_2'} \mathsf{V}$: Prover sends $z_1' \leftarrow r_1 - cc' \bmod p$, $z_2' \leftarrow r_2 \cdot z^{c'} \bmod p$.

V: Verifier checks that $a' = H^{z_1'} \cdot e(g, z_2') a^{c'}$.

B.2 Components for the Second Proof of Knowledge

Σ-PROTOCOL FOR PoK$\{(\sigma, v) : e(V, y) = e(V, g)^{-\sigma} e(g, g)^v\}$

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_{\mathrm{T}}$, instance $(V, y) \in \mathbb{G}_1^2$.

Prover's Input: $\sigma \in \{1, \ldots, N\}$, $v \in \mathbb{Z}_p$ s.t. $e(V, y) = e(V, g)^{-\sigma} e(g, g)^v$.

$\mathsf{P} \xrightarrow{a} \mathsf{V}$: Prover picks $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$ and sends $a \leftarrow e(V, g)^{-r_1} e(g, g)^{r_2}$.

$\mathsf{P} \xleftarrow{c} \mathsf{V}$: Verifier sends a random challenge $c \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z_1, z_2} \mathsf{V}$: Prover sends $z_1 \leftarrow r_1 - \sigma c \bmod p$ and $z_2 \leftarrow r_2 - vc \bmod p$.

V: Verifier checks that

$$a \stackrel{?}{=} e(V, y)^c \cdot e(V, g)^{-z_1} \cdot e(g, g)^{z_2} \; . \tag{1}$$

Σ-PROTOCOL FOR COMMITMENT RELATIONSHIP

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_{\mathrm{T}}$, instance $(V, y, a) \in \mathbb{G}_1^2 \times \mathbb{G}_{\mathrm{T}}$.

Prover's Input: $(c, (z_1, z_2)) \in \mathbb{Z}_p \times \mathbb{Z}_p^2$ such that Equation (1) holds.

$\mathsf{P} \xrightarrow{a'} \mathsf{V}$: Prover picks $r_1, r_2, r_3 \xleftarrow{\$} \mathbb{Z}_p$ and sends $a' = e(V, y)^{r_1} \cdot e(V, g)^{-r_2} \cdot e(g, g)^{r_3}$.

$\mathsf{P} \xleftarrow{c'} \mathsf{V}$: Verifier sends a random challenge $c' \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z_1', z_2', z_3'} \mathsf{V}$: Prover sends $z_1' \leftarrow r_1 - cc' \bmod p$, $z_2' \leftarrow r_2 - z_1 c' \bmod p$ and $z_3' \leftarrow r_3 - z_2 c' \bmod p$.

V: Verifier checks that $a' \stackrel{?}{=} a^{c'} \cdot e(V, y)^{z_1'} \cdot e(V, g)^{-z_2'} \cdot e(g, g)^{z_3'}$.

B.3 Components for the Third Proof of Membership

Although the third proof is a proof of membership, we use the same construction for the (stronger) proof of knowledge.

Σ-PROTOCOL FOR $\mathrm{PoM}\{(h) : H = e(g,h) \wedge W = e(h,V)\}$

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathrm{T}$, instance $(V, H, W) \in \mathbb{G}_1 \times \mathbb{G}_\mathrm{T}^2$.

Prover's Input: $h \in \mathbb{G}_1$ such that $H = e(g,h)$ and $W = e(h,V)$.

$\mathsf{P} \xrightarrow{a_1,a_2} \mathsf{V}:$ Prover chooses $r \xleftarrow{\$} \mathbb{G}_1$ and sends $a_1 \leftarrow e(g,r)$ and $a_2 \leftarrow e(r,V)$.

$\mathsf{P} \xleftarrow{c} \mathsf{V}:$ Verifier sends a random challenge $c \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z} \mathsf{V}:$ Prover sends $z \leftarrow r \cdot h^{-c}$.

$\mathsf{V}:$ Verifier checks

$$a_1 \stackrel{?}{=} e(g,z) \cdot H^c \quad \text{and} \quad a_2 \stackrel{?}{=} e(z,V) \cdot W^c \ . \tag{2}$$

SIGMA PROTOCOL FOR COMMITMENT RELATIONSHIP

Common Input: Group parameters for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathrm{T}$, instance $(V, H, W, a_1, a_2) \in \mathbb{G}_1 \times \mathbb{G}_\mathrm{T}^4$,

Prover's Input: $(c, z) \in \mathbb{Z}_p \times \mathbb{G}_1$ such that Equation (2) holds.

$\mathsf{P} \xrightarrow{a_1',a_2'} \mathsf{V}:$ Prover picks $r_1 \xleftarrow{\$} \mathbb{Z}_p$ and $r_2 \xleftarrow{\$} \mathbb{G}_1$, and sends $a_1' \leftarrow e(g,r_2) \cdot H^{r_1}$ and $a_2' \leftarrow e(r_2,V) \cdot W^{r_1}$.

$\mathsf{P} \xleftarrow{c'} \mathsf{V}:$ Verifier sends a random challenge $c' \xleftarrow{\$} \mathbb{Z}_p$.

$\mathsf{P} \xrightarrow{z_1',z_2'} \mathsf{V}:$ Prover sends $z_1' \leftarrow r_1 - cc' \bmod p$ and $z_2' \leftarrow r_2 \cdot z^{-c'}$.

$\mathsf{V}:$ Verifier checks that $a_1' \stackrel{?}{=} e(g,z_2') \cdot H^{z_1'} \cdot a_1^{c'}$ and $a_2' \stackrel{?}{=} e(z_2',V) \cdot H^{z_1'} \cdot a_2^{c'}$.

C PDDH in Generic Groups

We build confidence in our new PDDH assumption by showing its hardness in generic bilinear groups [Sho97]. In fact, we give a computational lower bound for a new problem that we call the *Vector General Diffie-Hellman Exponent* (VGDHE) problem, and that contains the PDDH problem as a special case. The VGDHE problem is an extension of the General Diffie-Hellman Exponent problem introduced by Boneh, Boyen and Goh [BBG05] where the adversary has to distinguish a vector of group elements from random, rather than a single element. Our proof in the generic group model is very similar to that of [BBG05], but is included here for completeness.

Let p be the prime group order and let $n \in \mathbb{N}$. Let $P, Q, F \subset \mathbb{Z}_p[X_1, \ldots, X_n]$ be sets of polynomials in variables $X_1, \ldots, X_n$. For $g \in \mathbb{G}_1$ and $x_1, \ldots, x_n \in \mathbb{Z}_p$ let $g^{P(x_1,\ldots,x_n)}$ denote the vector $\left(g^{p_1(x_1,\ldots,x_n)}, \ldots, g^{p_{|P|}(x_1,\ldots,x_n)}\right) \in \mathbb{G}_1^{|P|}$ where $P = \{p_1, \ldots, p_{|P|}\}$. The vectors $e(g,g)^{Q(x_1,\ldots,x_n)}$ and $e(g,g)^{F(x_1,\ldots,x_n)}$ are defined analogously. Algorithm A has advantage ϵ in solving the (P,Q,F)-VGDHE problem in $(\mathbb{G}_1, \mathbb{G}_\mathrm{T})$ if

$$\Big| \Pr\left[\mathsf{A}\left(g^{P(x_1,\ldots,x_n)}\ ,\ e(g,g)^{Q(x_1,\ldots,x_n)}\ ,\ e(g,g)^{F(x_1,\ldots,x_n)}\ ,\ T\right) = 1\right]$$
$$- \Pr\left[\mathsf{A}\left(g^{P(x_1,\ldots,x_n)}\ ,\ e(g,g)^{Q(x_1,\ldots,x_n)}\ ,\ T\ ,\ e(g,g)^{F(x_1,\ldots,x_n)}\right) = 1\right]\Big| \ > \ \epsilon\ ,$$

where the probability is taken over the random choices of $g \xleftarrow{\$} \mathbb{G}_1$, $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$, and $T \xleftarrow{\$} \mathbb{G}_{\mathrm{T}}^{|F|}$. We extend the independence definition of [BBG05] to the case that $|F| > 1$.

Definition C.1 Let $P, Q, F \subset \mathbb{Z}_p[X_1, \ldots, X_n]$ be sets of polynomials such that $P = \{p_1, \ldots, p_{|P|}\}$, $Q = \{q_1, \ldots, q_{|Q|}\}$, $F = \{f_1, \ldots, f_{|F|}\}$. We say that F is independent of P, Q if there does not exist a non-trivial (i.e., not all zeroes) assignment for the coefficients $\{a_{i,j}\}_{i,j=1}^{|P|}, \{b_i\}_{i=1}^{|Q|}, \{c_i\}_{i=1}^{|F|} \in \mathbb{Z}_p$ such that

$$\sum_{i=1}^{|P|} a_{i,j} p_i p_j + \sum_{i=1}^{|Q|} b_i q_i + \sum_{i=1}^{|F|} c_i f_i \;=\; 0 \bmod p \;.$$

The degree of a term $cX_1^{d_1} \cdots X_n^{d_n}$ is $d = d_1 + \ldots + d_n$; the degree of a polynomial $p \in \mathbb{Z}_p[X_1, \ldots, X_n]$ is the maximum of the degrees of all its terms; and the degree $\deg(P)$ of a set of polynomials $P \subset \mathbb{Z}_p[X_1, \ldots, X_n]$ is the maximum of the degrees of all its elements.

In the generic group model, an adversary A sees group elements only through an encoding as unique random strings. Let $\chi : \mathbb{Z}_p \to \{0,1\}^\ell$ be a function that maps $x \in \mathbb{Z}_p$ to the string representation $\chi(x)$ of $g^x \in \mathbb{G}_1$. Likewise, let $\xi : \mathbb{Z}_p \to \{0,1\}^\ell$ be such that $\xi(x)$ is the string representation of $e(g,g)^x$. The adversary has access to oracles for computing the group operations in $\mathbb{G}_1$ and $\mathbb{G}_{\mathrm{T}}$, and for the pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_{\mathrm{T}}$.

Theorem C.2 Let $P, Q, F \subset \mathbb{Z}_p[X_1, \ldots, X_n]$ as defined above with $d_P = \deg(P)$, $d_Q = \deg(Q)$, and $d_F = \deg(F)$. If A makes a total of q queries to its oracles, then its advantage in solving the VGDHE problem is at most

$$\epsilon \;\leq\; \frac{(|P| + |Q| + 2|F| + q)^2 \cdot d}{p} \;,$$

where $d = \max(2d_P, d_Q, d_F, 1)$.

Proof: Consider the algorithm B that provides an execution environment for A as follows. It maintains two lists of pairs

$$L_1 \;=\; \{(p_i, \chi_i) \;:\; i = 1, \ldots, l_1\} \;, \qquad L_{\mathrm{T}} \;=\; \{(q_i, \xi_i) \;:\; i = 1, \ldots, l_{\mathrm{T}}\} \;,$$

Initially, L_1 contains $|P|$ pairs (p_i, χ_i) where $\{p_1, \ldots, p_{|P|}\} = P$ and $\chi_1, \ldots, \chi_{|P|}$ are unique random ℓ-bit strings. The list L_{T} contains polynomials not only in the n variables $X_1, \ldots, X_n$, but also in $2|F|$ additional variables $Y_{0,1}, \ldots, Y_{0,|F|}, Y_{1,1}, \ldots, Y_{1,|F|}$. Initially it contains the $|Q| + 2|F|$ pairs (q_i, ξ_i) where $\{q_1, \ldots, q_{|Q|}\} = Q$, where $q_{|Q|+i} = Y_{0,i}$ for $1 \leq i \leq |F|$, and where $q_{|Q|+|F|+i} = Y_{1,i}$ for $1 \leq i \leq |F|$. Here also, the ξ_i are unique random ℓ-bit strings. At any point in the game we will have that $l_1 + l_{\mathrm{T}} \leq |P| + |Q| + 2|F| + q$, where q is the number of A's oracle queries.

We assume that A only queries its oracles with element representations that were either part of its input or that it obtained through previous oracle queries. This is reasonable because its probability of "predicting" an element encoding can be made arbitrarily small by increasing ℓ. Algorithm B responds to A's oracle queries as follows.

Multiplication in $\mathbb{G}_1$. On query (χ_i, χ_j), B looks up the pairs $(p_i, \chi_i), (p_j, \chi_j) \in L_1$. If there exists a pair $(p_k, \chi_k) \in L_1$ such that $p_i + p_j = p_k \bmod p$, then B returns χ_k. Otherwise, it increases l_1 and adds a pair (p_{l_1}, χ_{l_1}) to L_1 where $p_{l_1} = p_i + p_j \bmod p$ and $\chi_{l_1} \xleftarrow{\$} \{0,1\}^\ell \setminus \{\chi_1, \ldots, \chi_{l_1 - 1}\}$.

Multiplication in $\mathbb{G}_{\mathrm{T}}$. These queries are treated analogously, but using list L_{T} instead of L_1.

Pairing. On query (χ_i, χ_j), B looks up the pairs $(p_i, \chi_i), (p_j, \chi_j) \in L_1$. If there exists a pair $(q_k, \xi_k) \in L_{\mathrm{T}}$ such that $p_i \cdot p_j = q_k \bmod p$, then B returns ξ_k. Otherwise, it increases l_{T} and adds a pair $(q_{l_{\mathrm{T}}}, \xi_{l_{\mathrm{T}}})$ to L_{T} where $q_{l_{\mathrm{T}}} = p_i \cdot p_j \bmod p$ and $\xi_{l_{\mathrm{T}}} \xleftarrow{\$} \{0,1\}^\ell \setminus \{\xi_1, \ldots, \xi_{l_{\mathrm{T}}-1}\}$.

After q such queries, A outputs a bit b'. Now B chooses $b \xleftarrow{\$} \{0,1\}$ and sets $Y_{1-b,i} \leftarrow f_i(X_1, \ldots, X_n)$ for $1 \leq i \leq |F|$. Let BAD_1 be the event that after this assignment L_1 contains distinct pairs $(p_{i^*}, \chi_{i^*}), (p_{j^*}, \chi_{j^*})$ such that $p_{i^*} = p_{j^*} \bmod p$, or distinct pairs $(q_{i^*}, \xi_{i^*}), (q_{j^*}, \xi_{j^*}) \in L_{\mathrm{T}}$ such that $q_{i^*} = q_{j^*} \bmod p$. The first is clearly impossible because of the way B handles multiplication queries in $\mathbb{G}_1$ and because p_{i^*}, p_{j^*} do not contain any terms in $Y_{i,j}$. The second also turns out to be impossible, but this requires a bit more explanation. From the way that B handles pairing queries and multiplication queries in $\mathbb{G}_{\mathrm{T}}$ one can see that $q_{i^*} - q_{j^*}$ can be written as

$$\sum_{i,j=1}^{|P|} a_{i,j} p_{i,j} + \sum_{i=1}^{|Q|} b_i q_i + \sum_{i=1}^{|F|} c_i f_i + \sum_{i=1}^{|F|} d_i Y_{b,i}$$

for some constants $a_{i,j}, b_i, c_i, d_i$. For this polynomial to be identically zero, it has to hold that

$$\sum_{i,j=1}^{|P|} a_{i,j} p_i p_j + \sum_{i=1}^{|Q|} b_i q_i + \sum_{i=1}^{|F|} c_i f_i = 0 \bmod p$$

because neither of $\{p_i\}_{i=1}^{|P|}$, $\{q_i\}_{i=1}^{|Q|}$, or $\{f_i\}_{i=1}^{|F|}$ has terms in $Y_{b,j}$. This however contradicts the independence of F of P, Q.

Next, B chooses $x_1, \ldots, x_n, y_{b,1}, \ldots, y_{b,|F|} \xleftarrow{\$} \mathbb{Z}_p$. Let BAD_2 denote the event that this choice causes at least one "collision" in L_1 or L_{T}, meaning that

$$p_{i^*}(x_1, \ldots, x_n) - p_{j^*}(x_1, \ldots, x_n) \;=\; 0 \bmod p \tag{3}$$

for some $1 \leq i^* < j^* \leq l_1$, or that

$$q_{i^*}(x_1, \ldots, x_n, y_{b,1}, \ldots, y_{b,|F|}) - q_{j^*}(x_1, \ldots, x_n, y_{b,1}, \ldots, y_{b,|F|}) \;=\; 0 \bmod p \tag{4}$$

for some $1 \leq i^* < j^* \leq l_{\mathrm{T}}$. Here, we rewrote equations q_i as polynomials in $X_1, \ldots, X_n, Y_{b,1}, \ldots, Y_{b,|F|}$ after the assignment of $Y_{1-b,i} \leftarrow f_i(X_1, \ldots, X_n)$. If BAD_2 occurs, then B's simulation of A's environment is incorrect, because it returned two different encodings χ_{i^*}, χ_{j^*} (or ξ_{i^*}, ξ_{j^*}) for the same element. We therefore have to bound the probability that BAD_2 occurs.

We already argued that neither of Equations (3) or (4) is the zero polynomial, so the probability of hitting a root when choosing a random assignment is bounded from above by the Schwartz-Zippel theorem [Sch80] by the degree of the polynomial divided by p. For Equation (3) the degree is at most d_P, while for Equation (4) it is at most $d = \max(2d_P, d_Q, d_F, 1)$, so the probability of hitting a root for any of the equations is at most

$$\begin{aligned} \Pr[\,\text{BAD}_2\,] \;&\leq\; \binom{l_1}{2}\frac{d_P}{p} + \binom{l_{\mathrm{T}}}{2}\frac{d}{p} \\ &\leq\; \frac{(|P| + |Q| + 2|F| + q)^2 \cdot d}{2p}\,. \end{aligned}$$

Above, we used the facts that $d_P \leq d$, that $l_1 + l_{\mathrm{T}} \leq |P| + |Q| + 2|F| + q$, that $\binom{a+b}{2} \leq \binom{a}{2} + \binom{b}{2}$, and that $\binom{a}{2} \leq \frac{a^2}{2}$.

If event $\textsc{bad}_2$ does not occur, then B's simulation of A's environment is perfect. It is clear that in this case the probability that $b' = b$ is $1/2$, since b was chosen only after A output b'. We therefore have that

$$\begin{aligned}
\Pr\left[\, b' = 1 \;:\; b = 1 \,\right] &= \Pr\left[\, b' = 1 \;:\; b = 1 \wedge \neg\textsc{bad}_2 \,\right] \cdot \Pr\left[\, \neg\textsc{bad}_2 \,\right] \\
&\quad + \Pr\left[\, b' = 1 \;:\; b = 1 \wedge \textsc{bad}_2 \,\right] \cdot \Pr\left[\, \textsc{bad}_2 \,\right] \\
&\leq \Pr\left[\, b' = 1 \;:\; b = 1 \wedge \neg\textsc{bad}_2 \,\right] + \Pr\left[\, \textsc{bad}_2 \,\right] \\
&\leq \frac{1}{2} + \frac{(|P| + |Q| + 2|F| + q)^2 \cdot d}{2p} \,.
\end{aligned}$$

Likewise, we have that

$$\begin{aligned}
\Pr\left[\, b' = 1 \;:\; b = 0 \,\right] &= 1 - \Pr\left[\, b' = 0 \;:\; b = 0 \,\right] \\
&\geq \frac{1}{2} - \frac{(|P| + |Q| + 2|F| + q)^2 \cdot d}{2p} \,,
\end{aligned}$$

so the advantage of A is bounded by

$$\left| \Pr\left[\, b' = 1 \;:\; b = 1 \,\right] - \Pr\left[\, b' = 1 \;:\; b = 0 \,\right] \right| \;\leq\; \frac{(|P| + |Q| + 2|F| + q)^2 \cdot d}{p}$$

from which the theorem follows. ∎