Practical Adaptive Oblivious Transfer from Simple Assumptions

Matthew Green* and Susan Hohenberger**

Johns Hopkins University
{mgreen,susan}@cs.jhu.edu

Abstract. In an adaptive oblivious transfer (OT) protocol, a sender commits to a database of messages and then repeatedly interacts with a receiver in such a way that the receiver obtains one message per interaction of his choice (and nothing more) while the sender learns nothing about any of the choices. Recently, there has been significant effort to design practical adaptive OT schemes and to use these protocols as a building block for larger database applications. To be well suited for these applications, the underlying OT protocol should: (1) support an efficient initialization phase where *one* commitment can support an arbitrary number of receivers who are guaranteed of having the same view of the database, (2) execute transfers in time independent of the size of the database, and (3) satisfy a strong notion of security under a simple assumption in the standard model.

We present the first adaptive OT protocol simultaneously satisfying these requirements. The sole complexity assumption required is that given (g, g^a, g^b, g^c, Q), where g generates a bilinear group of prime order p and a, b, c are selected randomly from $\mathbb{Z}_p$, it is hard to decide if $Q = g^{abc}$. All prior protocols in the standard model either do not meet our efficiency requirements or require dynamic "q-based" assumptions.

Our construction makes an important change to the established "assisted decryption" technique for designing adaptive OT. As in prior works, the sender commits to a database of n messages by publishing an encryption of each message and a signature on each encryption. Then, each transfer phase can be executed in time *independent* of n as the receiver blinds one of the encryptions and proves knowledge of the blinding factors and a signature on this encryption, after which the sender helps the receiver decrypt the chosen ciphertext. One of the main obstacles to designing an adaptive OT scheme from a simple assumption is realizing a suitable signature for this purpose (i.e., enabling signatures on group elements in a manner that later allows for efficient proofs.) We make the observation that a secure signature scheme is not necessary for this paradigm, provided that signatures can only be forged in certain ways. We then show how to efficiently integrate an insecure signature into a secure adaptive OT construction.

* Supported by NSF CNS-1010928. This publication was also made possible by Grant Number HHS 90TR0003/01. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS.

** Supported by NSF CAREER CNS-1053886, a Microsoft New Faculty Fellowship and a Google Research Award.

1 Introduction

Oblivious transfer OT [35, 39] is a two-party protocol, where a Sender with messages $M_1, \ldots, M_N$ and a Receiver with indices $\sigma_1, \ldots, \sigma_k \in [1, N]$ interact in such a way that at the end the Receiver obtains $M_{\sigma_1}, \ldots, M_{\sigma_k}$ without learning anything about the other messages and the Sender does not learn anything about the choices $\sigma_1, \ldots, \sigma_k$. In the *adaptive* OT setting [33], the Receiver may obtain $M_{\sigma_{i-1}}$ before deciding on σ_i [33].

Our Goals. Adaptive OT is an interesting primitive. Like non-adaptive OT, it is a key building block for secure multi-party computation [40, 19, 28]. More practically, it captures the way an oblivious medical, financial or patent database would be accessed. Recently, there has been a focus on designing practical, privacy-preserving databases with access controls [15, 8] or pricing mechanisms [36] based on adaptive OT. Unfortunately, researchers trying to design more-complex systems on top of current adaptive OT protocols do not have any ideal choices. For a database with N messages supporting U Receivers with security parameter λ, such a protocol must be:

1. Extremely efficient, even when N, the database size, is large. In particular, the cost to transfer one message to one Receiver should depend only on the security parameter and not on N. I.e., a Receiver should not have to do work proportional to the size of the database to download one file. (This rules out a number of naive approaches as discussed below.)
2. Furthermore, since few databases serve only one user, it should be possible to extend the protocol to the case where there are *many* Receivers, each of whom receives a consistent view of the database. In particular, the ideal situation, which we achieve in this work, is to have a *non-interactive* initialization phase, where the Sender can do $O(\lambda N)$ work to form a commitment that can then be used for an arbitrary number of receivers. Several prior works (e.g., [10, 22, 27, 36]) support a relatively efficient initialization phase with $O(\lambda(N + U))$ total work. By adding a CRS and making some modifications, this can likely be reduced to $O(\lambda N)$ (although the complexity assumptions will still be an issue.) What one wishes to avoid, however, is an initialization phase that requires $O(\lambda NU)$ total work. I.e., the sender should not have to set up a *unique* database containing *all* of its files for *each* of its users. (This also rules out some basic approaches.)
3. Finally, since this protocol is designed to be a building block of larger applications, it is critical that it be a solid one. In particular, it should satisfy a strong notion of security (i.e., full-simulatability or UC) under a mild complexity assumption in the standard model. Unfortunately, while sufficiently practical protocols exist, they either require random oracles [10, 22], dynamic[1] assumptions [10, 22, 27, 36] or interactive assumptions [38].

Thus, a new construction based on new techniques is needed.

[1] These are also called *parametric* or *q-based* assumptions. An example is q-Strong Diffie-Hellman [3] (q-SDH): given $(g, g^x, g^{x^2}, g^{x^3}, \ldots, g^{x^q})$, where g generates a group

From Non-Adaptive to Adaptive OT for Single Receivers. Since it is known how to build non-adaptive OT protocols based on simple assumptions [21, 32, 34] such as Decisional Diffie-Hellman and Quadratic Residuosity, it is natural to ask why constructing adaptive protocols has proven so difficult. Given any fully-simulatable 1-out-of-N non-adaptive OT protocol, one can build a fully-simulatable k-out-of-N adaptive OT protocol for a *single* Receiver by sequentially executing k instances of the non-adaptive protocol and, before each execution, having the sender prove in zero-knowledge that the sequence of N messages used in execution i is the same as the sequence of N messages used in execution $i-1$ [10]. Unfortunately, for security parameter λ, this protocol requires a total of $O(Nk\lambda)$ work to transfer k messages for (only) *one* Receiver and is thus impractical for any application involving large databases.

Thus, when Camenisch, Neven and shelat [10] began to reinvestigate this problem in 2007, they stressed that the real challenge was to build an OT scheme where the sender makes an initial commitment to the database (which is assumed to be broadcast to all receivers), and then the two parties only exchange a *constant number* of group elements per transfer.

Our Contributions. We present an efficient, adaptive oblivious transfer protocol which is fully-simulatable under a simple, static assumption. The sole complexity assumption required is that given (g, g^a, g^b, g^c, Q), where g generates a bilinear group of prime order p and a, b, c are selected randomly from $\mathbb{Z}_p$, it is hard to decide if $Q = g^{abc}$. This assumption called *Decisional 3-Party Diffie-Hellman* has been used in prior works [31, 5, 25]. Our protocol is practical, although more costly than the very efficient Camenisch et al. protocol [10] by a constant factor. The database commitment in our scheme requires roughly $(9 + 7N)$ group elements, whereas the commitment in [10] required roughly $(3 + 2N)$ group elements. By including the mild Decision Linear assumption [4], we can efficiently make this initialization phase *non-interactive* as we discuss in Section 4.

Our construction introduces a twist on the *assisted decryption* approach to OT design, where the underlying signatures need not be existentially unforgeable provided that certain forgeries are not permitted. As we discuss, these techniques may be useful in simplifying the complexity assumptions in schemes beyond OT such as F-signatures and anonymous credentials [1].

Intuition behind our $\mathsf{OT}^N_{k\times 1}$ Construction. As with most previous $\mathsf{OT}^N_{k\times 1}$ constructions, our construction uses a technique known as *assisted decryption*. For $i = 1$ to N, the Sender commits to his database by encrypting each message as $C_i = \mathrm{Enc}(M_i)$, and publishes a public key and ciphertexts $(pk, C_1, \ldots, C_N)$. The Receiver then checks that each ciphertext is well-formed. To obtain a message, the Sender and Receiver engage in a *blind* decryption protocol, i.e., an interac-

---

of prime order p and x is a random value in $\mathbb{Z}_p$, it is hard to compute $(g^{1/(x+c)}, c)$ for any $c \in \mathbb{Z}_p^*$. Typically, when q-SDH is used as the foundation of an adaptive OT scheme, q must dynamically adjust to the number of files in the database. Thus, the assumption required actually changes based on how the protocol is used.

Protocol	Initialization Cost	Transfer Cost	Assumption	Security Defn
Folklore	$\cdot$	$O(\lambda N)$	general assumptions	Full Sim
KN [29]	$O(\lambda(N+U))$	$O(\lambda N)$	Decisional nth Residuosity/DDH	Full Sim
NP [33]	$\cdot$	$O(\lambda \lg(N))$	DDH + OT_1^2	Half Sim
KNP [30]	$O(\lambda NU)$	$O(\lambda)$	DDH	Full Sim*
CNS [10]	$O(\lambda(N+U))$	$O(\lambda)$	q-Power DDH + q-Strong DH	Full Sim
GH [22]	$O(\lambda(N+U))$	$O(\lambda)$	Decision Linear + q-Hidden LRSW	UC
JL [27]	$O(\lambda(N+U))$	$O(\lambda)$	Comp. Dec. Residuosity + q-DDHI	Full Sim
RKP [36]	$O(\lambda(N+U))$	$O(\lambda)$	DLIN + q-Hidden SDH + q-TDH	UC
§3.2	$O(\lambda(N+U))$	$O(\lambda)$	Decision 3-Party DH	Full Sim
§4	$O(\lambda N)$	$O(\lambda)$	Decision 3-Party DH + DLIN	Full Sim

Fig. 1. Survey of adaptive k-out-of-N Oblivious Transfer protocols secure in the standard model. Let λ be the security parameter, N the size of the database and U the number of receivers. The horizontal lines separate the schemes into efficiency categories, which improve as one scans down the table. While the least efficient categories can be realized using assumptions such as DDH, all prior attempts to achieve practicality have required a dynamic q-based complexity assumption. A $*$ denotes the construction meets a strictly weaker notion than the standard used in the other works.

tive protocol in which the Sender does not view the ciphertext he decrypts, but where the Receiver is convinced that decryption was done correctly.

The difficulty here is to prevent the Receiver from abusing the decryption protocol, e.g., by requesting decryptions of ciphertexts which were either not produced by the Sender or have been mauled. The folklore solution is to have the Receiver provide a proof that his request corresponds to $C_1 \vee C_2 \vee ... \vee C_N$. Of course, the cost of each transfer is now dependent on the total database size and thus this solution is no (asymptotically) better than the trivial solution mentioned above.

In Eurocrypt 2007, Camenisch, Neven and shelat [10] were the first to propose a method for executing "assisted decryption" efficiently. The sender signed each ciphertext value. The receiver was required to prove knowledge of a corresponding signature before the sender would assist him in decrypting a ciphertext. This clever approach reduced the $O(N\lambda)$ work per transfer required above, to only $O(\lambda)$ work, where λ is a security parameter.

More specifically, Camenisch, Neven and shelat [10] used a deterministic encryption scheme and a signature with a particular structure: for $pk = (g, g^x, H = e(g,h))$ and $sk = h$, let $C_i = \left(g^{\frac{1}{x+i}}, M_i \cdot e(g,h)^{\frac{1}{x+i}}\right)$. Recall that $g^{1/(x+i)}$ is a weak Boneh-Boyen signature [3] on i under g^x, and here only a polynomial number of "messages" (1 to N) are signed. While this scheme supports an elegant and efficient blind decryption protocol, it also requires strong q-based assumptions for both the indistinguishability of the ciphertexts as well as the unforgeability of the weak Boneh-Boyen signature. It is based on the q-Strong Diffie-Hellman and the q-Power Decisional Diffie-Hellman assumptions. The latter assumption states that given $(g, g^x, g^{x^2}, \ldots, g^{x^q}, H)$, where $g \in \mathbb{G}$ and $H \in \mathbb{G}_T$, it is hard to

distinguish the vector of elements $(H^x, H^{x^2}, \ldots, H^{x^q})$ from a vector of random elements in $\mathbb{G}_T$. In essence, the rigid structure of this (and all prior) constructions appear to require a similarly structured complexity assumption, which grows with the database size.

To move past this, we will "loosen" the structure of the ciphertext and signature enough to break the dependence on a structured assumption, but not so much as to ruin our ability to perform efficient proofs. Finding this balance proved highly non-trivial.

We now turn to how our construction works. We will encrypt using the Boneh-Boyen IBE [2], which has a public key $pk = (g, g_1 = g^a, g_2, h)$ and encrypts M as $(g^r, (g_1^i h)^r, e(g_1, g_2)^r M)$ for identity i and randomness $r \in \mathbb{Z}_p$. Then we will sign r. To do this, we need a standard model signature scheme from a simple assumption (which is itself somewhat rare.) We choose the *stateful* signatures of Hohenberger-Waters [26], which has a public key $pk = (g, g^b, u, v, d, w, z, h)$ and signs M as $(\sigma_1, \sigma_2, s, i)$ for state i and randomness $s, t \in \mathbb{Z}_p$, where $\sigma_1 = g^t,\ \sigma_2 = (u^M v^s d)^b (w^{\lceil \lg(i) \rceil} z^i h)^t$.

Attempt 1. Now, consider the construction obtained by combining the BB IBE, secure under Decisional Bilinear Diffie-Hellman, with the HW signature, secure under the Computational Diffie-Hellman assumption. Here we will encrypt the ith message using identity i (in the BB IBE) and state i (in the HW signature), with an extra u^r term to allow the Receiver to verify well-formedness:

$$g^r, \quad (g_1^i h)^r, \quad e(g_1, g_2)^r M, \quad g^t, \quad (u^r v^s d)^b (w^{\lceil \lg(i) \rceil} z^i h)^t, \quad u^r, \quad s$$

The Receiver can verify the well-formedness of the ith ciphertext $(c_1, \ldots, c_7)$ by checking that $e((g_1^i h), c_1) = e(g, c_2)$, $e(g, c_6) = e(c_1, u)$ and

$$e(g, c_5) = e(c_6 v^{c_7} d, g^b) e(w^{\lceil \lg(i) \rceil} z^i h, c_4).$$

It is important that the Receiver can verify the well-formedness of the ciphertext-signature pair, so that the simulator can properly extract the messages from a cheating Sender during the proof of security. It is a nice additional feature that our verification is public and non-interactive.

Attempt 2. However, the above construction still has a lot of problems. Recall that we want the Receiver to ask for a blind decryption of a given ciphertext by (somehow) sending in blinded portions of the ciphertext, proving that these portions are linked to r and proving that he knows a signature on r. Unfortunately, efficiently proving knowledge of the HW signature is problematic due to the $\lceil \lg(i) \rceil$ exponent. We could do this using a range proof [13, 9, 6, 7], however, this would require that we introduce stronger assumptions such as Strong RSA or q-Strong Diffie-Hellman. We could instead do a bit-by-bit proof, but this would severely hurt our efficiency. Instead, our solution is to drop this term entirely from the HW signature to obtain the ciphertext:

$$g^r, \quad (g_1^i h)^r, \quad e(g_1, g_2)^r M, \quad g^t, \quad (u^r v^s d)^b (z^i h)^t, \quad u^r, \quad s$$

One major issue is that dropping this term breaks the unforgeability of the signature scheme. Indeed, it is now possible for anyone to efficiently compute a signature on any index over a certain polynomial threshold as set in the proof of security. However, we specifically chose to encrypt with the Boneh-Boyen IBE for this purpose. We will set our parameters so that an adversary is free to forge signatures with states of $N+1$ and higher, where N is the size of our database. The key idea is that asking for decryptions on *different identities* will not help a malicious Receiver obtain information about the database messages; indeed, we could even hand him the secret key for those identities. This makes our proof much more efficient, however, there is still a large problem.

Attempt 3. To argue, in the proof of security, that no malicious Receiver can forge signatures on a state $i \in [1, N]$, we must *extract* this signature and its forgery message from the proof of knowledge. However, we cannot extract the "message" r from a cheating Receiver, because an honest Receiver will not know the randomness used in the ciphertexts created by the Sender. The most we can ask a Receiver to prove knowledge of is the signature on r comprised of (c_4, c_5, c_6, c_7) and the value g^r. Thus, we cannot extract from the Receiver a valid forgery of the HW signatures.

Moreover, we need a stronger security guarantee than HW signatures gave us (i.e., existential unforgeability under adaptive chosen message attack [20].) We need that: it is not only the case that an adversary cannot produce a pair (m, σ) for a new m; now the adversary cannot even produce the pair (g^m, σ) for a new m, where σ is a signature on m. Do such powerful signatures exist?

Indeed, this security notion was formalized as F-signatures by Belenkiy, Chase, Kohlweiss and Lysyanskaya [1], where they also required q-based complexity assumptions for their construction. Fortunately, we are able to show that the HW signatures (and our mangled version of them without the $w^{\lceil \lg(i) \rceil}$ term) remain F-unforgeable for $F(m) = g^m$ under a simple static assumption. (See [26] or the full version of this work [23] for the full details on HW; the mangled version is proven as part of the OT system in Section 3.3.) We tie both this version of the signature scheme and the Boneh-Boyen IBE together under a single assumption: given (g, g^a, g^b, g^c), it is hard to decide if $Q = g^{abc}$.

Comparison to Prior Work. Let us briefly compare our approach to prior works; see Figure 1 for more. As we mention above, Camenisch, Neven and shelat [10] gave the first efficient, fully-simulatable construction for adaptive (and non-adaptive) OT. It is secure in the standard model under the q-Strong Diffie-Hellman and the q-Power Decisional Diffie-Hellman assumptions. They also provided a scheme in the random oracle model from unique blind signatures.

Green and Hohenberger [21] provided an adaptive OT construction in the random oracle model based on the Decisional Bilinear Diffie-Hellman assumption, namely, that given (g, g^a, g^b, g^c, Q), it is hard to decide if $Q = e(g, g)^{abc}$. In their construction, the Sender encrypted each message i under identity i using a IBE system. Then they provided a blind key extraction protocol, where the Receiver could blindly obtain a secret key for any identity of her choice.

In the assisted decryption setting, Green and Hohenberger [22] took an approach similar to [10] to achieve UC security. It was based on the Decision Linear and q-Hidden LRSW assumptions, in the asymmetric setting. The latter assumption implies that DDH must hold in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

Jarecki and Liu [27] took an alternative view: for $pk = g^x$, let $C_i = M_i \cdot g^{1/(x+i)}$. Recall that $g^{1/(x+i)}$ is also the Dodis-Yampolskiy pseudorandom function on input i [18]. This essentially simplifies the Camenisch et al. construction and allows a fully-simulatable scheme based on the Composite Decisional Residuosity and q-Decisional Diffie-Hellman Inversion assumptions. The blind decryption protocol involves obliviously evaluating the PRF on input i, which requires some costly zero knowledge proofs. However, this protocol is interesting as the only efficient and fully-simulatable protocol that does not require bilinear groups.

Rial, Kohlweiss and Preneel [36] presented a *priced* version of UC-secure adaptive OT using the assisted decryption approach. In priced OT, the obliviousness property must hold, even though the items being sold may have unique prices. The scheme is secure in the standard model under the Decision Linear, q-Triple Diffie-Hellman, and q-Hidden Strong Diffie-Hellman assumptions.

Unfortunately, all of these constructions have a rigid structure and seem to require a structured complexity assumption. We show that this structure can be enforced, not on the message itself, but rather through the *identity* of the encryption and the *state* of the signature. This provides us with enough glue to keep the security of the scheme together without overdoing it.

Recently, Kurosawa and Nojima [29] and Chen, Chou and Hou [14] gave adaptive OT constructions which purported to improve the underlying complexity assumptions of the schemes above, but which actually resorted to $O(N\lambda)$ transfer cost. It was already known how to achieve this level of (in)efficiency from *general* assumptions, including those of [29, 14], by following the folklore method for building adaptive OT from any non-adaptive OT system, as described in [17, 10] and the opening of our introduction. Moreover, [14] is set in the random oracle model.

Very recently[2], Kurosawa, Nojima and Phong [30] gave an adaptive OT construction from DDH with $O(\lambda)$ transfers. However, their work has several technical issues. First, their construction does not satisfy the standard full simulation definition used in [10, 21, 22, 27, 36] and this work. In [30], if a receiver ever requests the same file twice (say, she downloads a patent one day, deletes it, then downloads it again a month later), then this can be detected by the sender. This is at odds with the full simulation definition where the adversarial sender is only told by the ideal functionality that a file has been requested and thus cannot detect a repeated download. Second, it is not obvious how to modify their construction to satisfy the full simulation notion. One approach might be to make the receiver stateful and store every file she ever requests. This has the obvious drawback of requiring permanent storage of the decrypted messages, which may not be practical and is not a requirement in other works. Moreover, sub-

[2] The work of [30] appeared after the initial posting of this work [23].

tle technical issues arise as to what the receiver sends during a repeated query round. Third, their construction requires a very expensive initialization procedure where the sender must transmit, then receive back and store $O(N\lambda)$ bits for *each* receiver. In contrast, all prior practical work [10, 21, 22, 27, 36] and our results only require that the sender publish and store *one* $O(N\lambda)$ bit database for *all* receivers.

Thus, we build on this body of prior work to present the first efficient scheme satisfying the standard notion of full simulation from a simple assumption in the standard model.

2 Technical Preliminaries

Bilinear Groups. Let $\mathsf{BMsetup}$ be an algorithm that, on input 1^κ, outputs the parameters for a bilinear mapping as $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e)$, where g generates $\mathbb{G}$, the groups $\mathbb{G}, \mathbb{G}_T$ have prime order $p \in \Theta(2^\kappa)$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Two algebraic properties required are that: (1) if g generates $\mathbb{G}$, then $e(g, g) \neq 1$ and (2) for all $a, b \in \mathbb{Z}_p$, it holds that $e(g^a, g^b) = e(g, g)^{ab}$.

Assumption 1 (Decisional 3-Party Diffie-Hellman (3DDH) [31, 5, 25]) *Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^\lambda)$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is $1/2$ plus an amount negligible in λ:*

$$\Pr[g, z_0 \leftarrow \mathbb{G}; a, b, c \leftarrow \mathbb{Z}_p; z_1 \leftarrow g^{abc}; d \leftarrow \{0, 1\}; d' \leftarrow \mathcal{A}(g, g^a, g^b, g^c, z_d) : d = d'].$$

Proofs of Knowledge. We use known zero-knowledge and witness indistinguishable techniques for proving statements about discrete logarithms and their natural extensions to proving statements about bilinear groups, such as (1) proof of knowledge of a discrete logarithm modulo a prime [37] and (2) proof of the disjunction or conjunction of any two statements [16]. These are typically interactive, 4-round protocols. We discuss further implementation details in the full version [23].

When referring to the proofs above, we will use the notation of Camenisch and Stadler [11]. For instance, $ZKPoK\{(x, h) : y = g^x \wedge H = e(y, h) \wedge (1 \leq x \leq n)\}$ denotes a zero-knowledge proof of knowledge of an integer x and a group element $h \in \mathbb{G}$ such that $y = g^x$ and $H = e(y, h)$ holds and $1 \leq x \leq n$. All values not enclosed in ()'s are assumed to be known to the verifier.

3 Adaptive Oblivious Transfer from a Simple Assumption

Adaptive Oblivious Transfer ($\mathsf{OT}^N_{k\times 1}$) is traditionally defined as a protocol conducted by a Sender and a single Receiver. In the following section we will formally define the protocol and its security requirements. As noted above, a primary application of $\mathsf{OT}^N_{k\times 1}$ is the construction of multi-user oblivious databases, and thus we must also consider the implications of a protocol involving $U \geq 1$ distinct

Receivers. In the full version [23], we present an alternative definition that captures this notion and describes the security and consistency properties involved in such an interaction.[3]

3.1 Definition of Adaptive k-out-of-N Oblivious Transfer ($\mathbf{OT}^N_{k\times 1}$) [33, 10]

An oblivious transfer scheme is a tuple of algorithms $(\mathsf{S_I}, \mathsf{R_I}, \mathsf{S_T}, \mathsf{R_T})$. During the initialization phase, the Sender and the Receiver conduct an interactive protocol, where the Sender runs $\mathsf{S_I}(M_1, \ldots, M_N)$ to obtain state value S_0, and the Receiver runs $\mathsf{R_I}()$ to obtain state value R_0. Next, for $1 \leq i \leq k$, the i^{th} transfer proceeds as follows: the Sender runs $\mathsf{S_T}(S_{i-1})$ to obtain state value S_i, and the Receiver runs $\mathsf{R_T}(R_{i-1}, \sigma_i)$ where $1 \leq \sigma_i \leq N$ is the index of the message to be received. The receiver obtains state information R_i and the message M'_{σ_i} or $\perp$ indicating failure.

Definition 1 (Full Simulation Security.). Consider the following experiments.[4]

Real experiment. In experiment $\mathbf{Real}_{\hat{\mathsf{S}},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma)$, the possibly cheating sender $\hat{\mathsf{S}}$ is given messages $(M_1, \ldots, M_N)$ as input and interacts with the possibly cheating receiver $\hat{\mathsf{R}}(\Sigma)$, where Σ is a selection algorithm that on input the full collection of messages thus far received, outputs the index σ_i of the next message to be queried. At the beginning of the experiment, both $\hat{\mathsf{S}}$ and $\hat{\mathsf{R}}$ output initial states (S_0, R_0). In the transfer phase, for $1 \leq i \leq k$ the sender computes $S_i \leftarrow \hat{\mathsf{S}}(S_{i-1})$, and the receiver computes $(R_i, M'_i) \leftarrow \hat{\mathsf{R}}(R_{i-1})$, where M'_i may or may not be equal to M_i. At the end of the k^{th} transfer the output of the experiment is (S_k, R_k).

We define the *honest* Sender S as one that runs $\mathsf{S_I}(M_1, \ldots, M_N)$ in the first phase, during each transfer runs $\mathsf{S_T}()$ and outputs $S_k = \varepsilon$ as its final output. The *honest* Receiver R runs $\mathsf{R_I}$ in the first phase, and $\mathsf{R_T}(R_{i-1}, \sigma_i)$ at the i^{th} transfer, and outputs $R_k = (M'_{\sigma_1}, \ldots, M'_{\sigma_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathsf{S}}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$ the possibly cheating sender algorithm $\hat{\mathsf{S}}'$ generates messages $(M^*_1, \ldots, M^*_N)$ and transmits them to a trusted party T. In the i^{th} round $\hat{\mathsf{S}}'$ sends a bit b_i to T; the possibly cheating receiver $\hat{\mathsf{R}}'(\Sigma)$ transmits σ^*_i to T. If $b_i = 1$ and $\sigma^*_i \in \{1, \ldots, N\}$ then T hands $M^*_{\sigma_i{}^*}$ to $\hat{\mathsf{R}}'$. If $b_i = 0$ then T hands $\perp$ to $\hat{\mathsf{R}}'$. After the k^{th} transfer the output of the experiment is (S_k, R_k).

---

[3] Indeed, a multi-receiver definition is necessary to achieve consistency with the oblivious access control schemes of Coull et al. [15] and Camenisch et al. [8].

[4] As in [10], we do not explicitly specify auxiliary input to the parties; this information can (and indeed must) be provided in order to achieve sequential composition.

Let $\ell(\cdot)$ be a polynomially-bounded function. We now define Sender and Receiver security.

Sender Security. An $\mathsf{OT}^N_{k\times 1}$ provides Sender security if for every real-world p.p.t. receiver $\hat{\mathsf{R}}$ there exists a p.p.t. ideal-world receiver $\hat{\mathsf{R}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \ldots, M_N)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma).$$

Receiver Security. $\mathsf{OT}^N_{k\times 1}$ provides Receiver security if for every real-world p.p.t. sender $\hat{\mathsf{S}}$ there exists a p.p.t. ideal-world sender $\hat{\mathsf{S}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \ldots, M_N)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \Sigma).$$

3.2 The Construction

Our $\mathsf{OT}^N_{k\times 1}$ protocol appears in Figure 2. This protocol follows the *assisted (or blind) decryption* paradigm pioneered by [10, 22, 27]. The Sender begins the OT protocol by encrypting each message in the database and publishing these values to the Receiver. The Receiver then checks that each ciphertext is well-formed. For each of k transfers, the two parties co-operatively execute a protocol following which (1) the Receiver obtains the decryption of at most one ciphertext, while (2) the Sender learns nothing about *which* ciphertext was decrypted. We require that the interactive decryption protocol run in time independent of the size of the database.

The encryption scheme that we use is a novel combination of the Boneh-Boyen IBE scheme [2] and a (insecure) version of the Hohenberger-Waters signatures [26]. We present methods for proving knowledge of such signatures and obtaining a blind decryption. Of course, in an adaptive OT scheme, the difficulty is always in getting all elements of the fully-simulatable proof of security to work out. There are many subtle details in basing the security for any database of size N under the one simple assumption that given (g, g^a, g^b, g^c), it is hard to decide if $Q = g^{abc}$.

Ciphertext Structure. In Figure 2, we reference a $\mathsf{VerifyCiphertext}$ algorithm for verifying the well-formedness of a ciphertext. Let us explain that now. The Sender's public parameters pk include $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e)$ and generators $(g_1, g_2, h, g_3, g_4, u, v, d) \in \mathbb{G}^8$. For message $M \in \mathbb{G}_T$, identity $j \in \mathbb{Z}_p$, and random values $r, s, t \in \mathbb{Z}_p$ we can express a ciphertext as:

$$C = \Big(g^r,\ (g_1^j h)^r,\ M \cdot e(g_1, g_2)^r,\ g^t,\ (u^r v^s d)^b (g_3^j h)^t,\ u^r,\ s\Big)$$

Given only pk, j, the $\mathsf{VerifyCiphertext}$ function validates that the ciphertext has this structure. We define it as follows.

$\mathsf{VerifyCiphertext}(pk, C, j)$. Parse C as $(c_1, \ldots, c_7)$ and pk to obtain g, g_1, h, g_3, g_4, u, v, d. This routine outputs 1 if and only if the following equalities hold:

$$\begin{aligned} e(g_1^j h, c_1) &= e\,(g, c_2) \quad \wedge \\ e\,(g, c_6) &= e\,(c_1, u) \quad \wedge \\ e\,(g, c_5) &= e\,(g_4, c_6 v^{c_7} d)\, e(c_4, g_3^j h) \end{aligned}$$

This function will always output 1 when input a properly-formed ciphertext.

$\underline{\mathsf{S_I}(M_1, \ldots, M_N)}$	$\underline{\mathsf{R_I}()}$
1. Select $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BMsetup}(1^\kappa)$ and $a, b \xleftarrow{\$} \mathbb{Z}_p$, choose $g_2, g_3, h, u, v, d \xleftarrow{\$} \mathbb{G}$ and set $g_1 \leftarrow g^a, g_4 \leftarrow g^b$. Let $pk = (\gamma, g_1, g_2, g_3, g_4, h, u, v, d)$ and $sk = (a, b)$.	
2. For $j = 1$ to N, select $r_j, s_j, t_j \xleftarrow{\$} \mathbb{Z}_p$ and set: $C_j \leftarrow [g^{r_j},\ (g_1^j h)^{r_j},\ M_j e(g_1, g_2)^{r_j},\ g^{t_j},\ (u^{r_j} v^{s_j} d)^b (g_3^j h)^{t_j},\ u^{r_j},\ s_j]$.	
3. Send $(pk, C_1, \ldots, C_N)$ to Receiver.	
4. Conduct $ZKPoK\{(a) : g_1 = g^a\}$.	
	5. Verify pk and the proof.[a] Check for $j = 1$ to N: $\mathsf{VerifyCiphertext}(pk, C_j, j) = 1$. If any check fails, output $\perp$.
Output $S_0 = (pk, sk)$.	Output $R_0 = (pk, C_1, \ldots, C_N)$.
$\underline{\mathsf{S_T}(S_{i-1})}$	$\underline{\mathsf{R_T}(R_{i-1}, \sigma_i)}$
	1. Parse C_{σ_i} as $(c_1, \ldots, c_7)$, select $x, y \xleftarrow{\$} \mathbb{Z}_p$ and compute $v_1 \leftarrow g^x c_1$.
	2. Send v_1 to Sender, and conduct: $WIPoK\{(\sigma_i, x, c_2, c_4, c_5, c_6, c_7) : e(v_1/g^x, (g_1^{\sigma_i} h)) = e(c_2, g) \wedge e(c_6, g) = e(v_1/g^x, u) \wedge e(c_5, g) = e(c_6 v^{c_7} d, g_4) e(c_4, g_3^{\sigma_i} h)\}$.
3. Set $R \leftarrow e(v_1, g_2^a)$.	
4. Send R to Receiver and conduct: $ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.	
	5. If the proof does not verify, output $\perp$. Else output $M'_{\sigma_i} \leftarrow \frac{c_3 \cdot e(g_1, g_2)^x}{R}$.
Output $S_i = S_{i-1}$.	Output $R_i = (R_{i-1}, M'_{\sigma_i})$.

[a] By verify pk, we mean check that γ contains parameters for a bilinear map, where p is prime and g generates $\mathbb{G}$ with order p. Also, verify that the remaining pk elements are members of $\mathbb{G}$.

Fig. 2. Our adaptive $\mathsf{OT}_{k\times 1}^N$ protocol. $\mathsf{VerifyCiphertext}$ is described above.

3.3 Security Analysis

We now show that the $\mathsf{OT}^N_{k\times 1}$ protocol above is sender-secure and receiver-secure in the full-simulation model under the Decisional 3-Party Diffie-Hellman assumption (3DDH). We will address Sender and Receiver security separately.

A note on the PoK protocols. For generality, our security proofs use the terms $\epsilon_{ZK}, \epsilon_{WI}$ to indicate the maximal advantage that every p.p.t. distinguisher has in distinguishing simulated ZKPoKs from real ones (*resp.* WI proofs on different witnesses). We additionally use ϵ_{Ext} to indicate the maximum probability that the extractor for a PoK fails (soundness). We propose to use four-round Schnorr proofs which are zero-knowledge/WI ($\epsilon_{WI} = \epsilon_{ZK} = 0$) and computationally sound under the Discrete Logarithm assumption (which is naturally implied by 3DDH). Our security proofs employ the knowledge extractors for these proofs-of-knowledge, which we will define as $\mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3$.[5]

Sender security. Given a (possibly cheating) real-world receiver $\hat{\mathsf{R}}$, we show how to construct an ideal-world receiver $\hat{\mathsf{R}}'$ such that all p.p.t. distinguishers have at most negligible advantage in distinguishing the distribution of an honest real-world sender S interacting with $\hat{\mathsf{R}}$ ($\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$) from that of $\hat{\mathsf{R}}'$ interacting with the honest ideal-world sender S' ($\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}$). Let us now describe the operation of $\hat{\mathsf{R}}'$, which runs $\hat{\mathsf{R}}$ internally, interacting with it in the role of the Sender:

1. To begin, $\hat{\mathsf{R}}'$ selects a random collection of messages $\bar{M}_1, \ldots, \bar{M}_N \xleftarrow{\$} \mathbb{G}_T$ and follows the S_I algorithm (from Figure 2) with these as input up to the point where it obtains $(pk, C_1, \ldots, C_N)$.
2. It sends $(pk, C_1, \ldots, C_N)$ to $\hat{\mathsf{R}}$ and then *simulates* the interactive proof $ZKPoK\{(a) : g_1 = g^a\}$. (Even though $\hat{\mathsf{R}}'$ knows $sk = a$, it ignores this value and simulate this proof step.)
3. For each of k transfers initiated by $\hat{\mathsf{R}}$,
   (a) $\hat{\mathsf{R}}'$ verifies the received WIPoK and uses the knowledge extractor E_2 to obtain the values $\sigma_i, x, y, c_1, c_2, c_3, c_4$ from it. $\hat{\mathsf{R}}'$ aborts and outputs error when E_2 fails.
   (b) When $\sigma_i \in [1, N]$, $\hat{\mathsf{R}}'$ queries the trusted party T to obtain M_{σ_i}, parses C_{σ_i} as $(c_1, \ldots, c_7)$ and responds with $R = \frac{c_3 e(g_1, g_2)^x}{M_{\sigma_i}}$ (if T returns $\perp$, $\hat{\mathsf{R}}'$ aborts the transfer). When $\sigma_i \notin [1, N]$, $\hat{\mathsf{R}}'$ follows the normal protocol. In both cases, $\hat{\mathsf{R}}'$ simulates $ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.
4. $\hat{\mathsf{R}}'$ uses $\hat{\mathsf{R}}$'s output as its own.

Theorem 2. *Let ϵ_{ZK} be the maximum advantage with which any p.p.t. algorithm distinguishes a simulated ZKPoK, and ϵ_{Ext} be the maximum probability that the extractor E_2 fails (with ϵ_{ZK} and ϵ_{Ext} both negligible in κ). If all p.p.t.*

[5] These correspond respectively to the proofs $ZKPoK\{(a) : g_1 = g^a\}$, $WIPoK\{(\sigma_i, x, y, z, c_4, c_5, c_6, c_7) : \ldots\}$, and $ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.

algorithms have negligible advantage $\leq \epsilon$ at solving the 3DDH problem, then:

$$\mathbf{Pr}\left[D(\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \right] - \mathbf{Pr}\left[D(\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \right] \leq (k+1)\epsilon_{ZK} + k\epsilon_{Ext} + N\epsilon\left(1 + \frac{p}{p-1}\right).$$

A proof of Theorem 2 is sketched in Appendix A.1 and detailed in [23].

Receiver Security. For any real-world cheating sender $\hat{\mathsf{S}}$ we can construct an ideal-world sender $\hat{\mathsf{S}}'$ such that all p.p.t. distinguishers have negligible advantage at distinguishing the distribution of the real and ideal experiments. Let us now describe the operation of $\hat{\mathsf{S}}'$, which runs $\hat{\mathsf{S}}$ internally, interacting with it in the role of the Receiver.

1. To begin, $\hat{\mathsf{S}}'$ runs the R_I algorithm, with the following modification: when $\hat{\mathsf{S}}$ proves knowledge of a, $\hat{\mathsf{S}}'$ uses the knowledge extractor E_1 to extract a, outputting error if the extractor fails. Otherwise, it has obtained the values $(pk, C_1, \ldots, C_N)$.
2. For $i = 1$ to N, $\hat{\mathsf{S}}'$ decrypts each of $\hat{\mathsf{S}}$'s ciphertexts $C_1, \ldots, C_N$ using the value a as a decryption key,[6] and sends the resulting $M_1^*, \ldots, M_N^*$ to the trusted party T.
3. Whenever T indicates to $\hat{\mathsf{S}}'$ that a transfer has been initiated, $\hat{\mathsf{S}}'$ runs the transfer protocol with $\hat{\mathsf{S}}$ on the fixed index 1. If the transfer succeeds, $\hat{\mathsf{S}}'$ returns the bit 1 (indicating success) to T, or 0 otherwise.
4. $\hat{\mathsf{S}}'$ uses $\hat{\mathsf{S}}$'s output as its own.

Theorem 3. *Let ϵ_{WI} be the maximum advantage that any p.p.t. algorithm has at distinguishing a WIPoK, and let ϵ_{Ext} be the maximum probability that the extractor E_1 fails. Then $\forall$ p.p.t. D:*

$$\mathbf{Pr}\left[D(\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \right] - \mathbf{Pr}\left[D(\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \right] \leq (k+1)\epsilon_{Ext} + k\epsilon_{WI}.$$

A proof of Theorem 3 is sketched in Appendix A.2 and detailed in [23].

4 Efficiently Supporting Multiple Receivers

Adaptive Oblivious Transfer ($\mathsf{OT}^N_{k\times 1}$) is traditionally defined as a protocol between a Sender and a single Receiver. However, the way it is typically used in

[6] Parse C_i as $(c_1, \ldots, c_7)$ and decrypt as $M_i^* = c_3/e(c_1, g_2^a)$. As noted in Section 4, one can modify the protocol so that the Sender conducts a PoK of the value g_2^a.

practical works such as Coull et al. [15] and Camenisch et al. [8] is that $U \geq 1$ distinct Receivers all interact with a single Sender.

Extending the full simulation definition to cover this explicitly is rather straightforward. We do so in the full version [23]. The main technical concern is that every Receiver should have the same view of the database. That is, if two Receivers make a request on index i and neither transfer resulted in an error, then those Receivers must have obtained the same message. In [23] we explain why our construction would satisfy such a notion – namely, that all Receivers share the same database and a Receiver does not accept a message unless the Sender can prove that it correctly corresponds to this database. For simplicity, we assume secure channels for the transfer phase.

Eliminating the $O(\lambda U)$ term. Interestingly, we can also improve the efficiency of our initialization protocol when multiple Receivers are present. In the current protocol of Figure 2, the Sender must conduct the proof of knowledge $ZKPoK\{(a) : g_1 = g^a\}$ with each Receiver. This can be accomplished using a very inexpensive interactive four-round proof in the standard model.

Fortunately even this minimal per-user initialization can be eliminated by assuming a Common Reference String shared by the Sender and all Receivers and using an NIZKPoK to broadcast this proof to all Receivers. To instantiate this proof, we suggest the efficient proof system of Groth and Sahai [24], which permits proofs of pairing-based statements under the Decision Linear assumption [4]. One wrinkle in this approach is that our proof of Receiver security assumes that the simulator can extract the trapdoor $a \in \mathbb{Z}_p$ from the ZKPoK, in order to decrypt the ciphertext vector $C_1, \ldots, C_N$. However, the knowledge extractor for the Groth-Sahai proof system is limited in that it can only extract elements of the bilinear image group $\mathbb{G}$. Fortunately for our purposes, the value $g_2^a \in \mathbb{G}$ can be used as an alternative trapdoor (see Section 3.3 for details). Thus when using the Groth-Sahai system we must re-write the proof as $NIZKPoK\{(g_2^a) : e(g_1, g_2) = e(g_2^a, g)\}$.[7]

5 Conclusions and Open Problems

We presented the first efficient, adaptive oblivious transfer protocol which is fully-simulatable under simple, static assumptions. Our protocol is practical and can be used as a building block in larger database applications, such as [15, 36, 8], as a step to reducing the overall assumptions on the system.

We leave open several interesting problems. First, we use standard zero-knowledge proof and extraction techniques which require rewinding, and thus, our scheme is not UC-secure. A natural question is whether one can obtain UC-security by replacing our interactive proofs with the non-interactive Groth-Sahai proofs [24]. Unfortunately, this is not an easy substitution. Our security proofs

[7] As mentioned by Groth and Sahai, statements of this form must be slightly re-written to enable full zero knowledge. The equivalent statement is $ZKPoK\{(g_2^a, g_1') : e(g_2^a, g)e(g_1', g_2^{-1}) = 1 \ \wedge \ e(g_1', g) = e(g_1, g)\}$.

use techniques from the Boneh-Boyen cryptosystem that depend fundamentally on the ability to extract *integers* from the Receiver's proof of knowledge during the Transfer phase. The Groth-Sahai proof system is only F-extractable, meaning that one can obtain only group elements from the extractor (even when the proof is over integer witnesses). One can easily substitute a bit-by-bit proof of each integer, but we would hope to identify a more practical approach.

It would be interesting to know if the observations about and manipulations of the Hohenberger-Waters signatures [26] identified in this work could be extended to applications such as anonymous credentials and electronic cash, where most efficient constructions still require random oracles or strong complexity assumptions. One of the main difficulties is that many interesting protocols require an F-signature together with an efficient range proof (i.e., method for proving in zero-knowledge that a committed value lies within a public range.) Currently, the only efficient techniques for doing the latter require either the Strong RSA assumption [13, 9, 6] or (more recently) the q-Strong Diffie-Hellman assumption [7, 12]. (Here q need only be linked to a security parameter, e.g., $q = 256$.) It would be interesting if range proofs under weaker assumptions could be devised.

References

1. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. *P*-signatures and noninteractive anonymous credentials. In *Theory of Cryptography Conference*, volume 4948 of LNCS, pages 356–374, 2008.
2. Dan Boneh and Xavier Boyen. Efficient selective-ID secure Identity-Based Encryption without random oracles. In *EUROCRYPT '04*, volume 3027 of LNCS, pages 223–238, 2004.
3. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT '04*, volume 3027 of LNCS, pages 382–400, 2004.
4. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO '04*, volume 3152 of LNCS, pages 45–55, 2004.
5. Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Advances in Cryptology – EUROCRYPT '06*, volume 4004 of LNCS, pages 573–592, 2006.
6. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT '00*, volume 1807 of LNCS, pages 431–444, 2000.
7. Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient protocols for set membership and range proofs. In *ASIACRYPT '08*, volume 5350 of LNCS, pages 234–252. Springer, 2008.
8. Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access controls. In *ACM CCS '09*, pages 131–140, 2009.
9. Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number n is the product of two safe primes. In *EUROCRYPT '99*, volume 1592 of LNCS, pages 107–122, 1999.
10. Jan Camenisch, Gregory Neven, and abhi shelat. Simulatable adaptive oblivious transfer. In *EUROCRYPT '07*, volume 4515 of LNCS, pages 573–590, 2007.
11. Jan Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *CRYPTO '97*, volume 1296 of LNCS, pages 410–424, 1997.

12. Rafik Chaabouni, Helger Lipmaa, and Abhi Shelat. Additive combinatorics and discrete logarithm based range protocols, 2009. Cryptology ePrint Archive: 2009/469. Available at `http://eprint.iacr.org/2009/469.pdf`.
13. Agnes Chan, Yair Frankel, and Yiannis Tsiounis. Easy come – easy go divisible cash. In *EUROCRYPT '98*, volume 1403 of LNCS, pages 561–575, 1998.
14. Yalin Chen, Jue-Sam Chou, and Xian-Wu Hou. A novel k-out-of-n oblivious transfer protocols based on bilinear pairings, 2010. Cryptology ePrint Archive: Report 2010/027.
15. Scott Coull, Matthew Green, and Susan Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In *Public Key Cryptography*, volume 5443 of LNCS, pages 501–520, 2009.
16. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, volume 839 of LNCS, pages 174–187, 1994.
17. Yevgeniy Dodis, Shai Halevi, and Tal Rabin. A cryptographic solution to a game theoretic problem. In *CRYPTO '00*, volume 1880 of LNCS, pages 112–130, 2000.
18. Yevgeniy Dodis and Aleksandr Yampolskiy. A Verifiable Random Function with Short Proofs an Keys. In *Public Key Cryptography '05*, volume 3386 of LNCS, pages 416–431, 2005.
19. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC '87*, pages 218–229, 1987.
20. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2), 1988.
21. Matthew Green and Susan Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *ASIACRYPT '07*, volume 4833 of LNCS, pages 265–282, 2007.
22. Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT '08*, volume 5350 of LNCS, pages 179–197, 2008.
23. Matthew Green and Susan Hohenberger. Practical adaptive oblivious transfer from simple assumptions, 2010. The full version of this work appears in the Cryptology ePrint Archive: Report 2010/109.
24. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT '08*, volume 4965 of LNCS, pages 415–432. Springer, 2008.
25. Susan Hohenberger, Guy N. Rothblum, abhi shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In *TCC '07*, volume 4392 of LNCS, pages 233–252, 2007.
26. Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In *EUROCRYPT '09*, volume 5479 of LNCS, pages 333–350, 2009.
27. Stanislaw Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In *TCC '09*, volume 5444 of LNCS, pages 577–594, 2009.
28. Joe Kilian. Founding cryptography on oblivious transfer. In *STOC '88*, pages 20–31, 1988.
29. Kaoru Kurosawa and Ryo Nojima. Simple adaptive oblivious transfer without random oracle. In *ASIACRYPT '09*, volume 5912 of LNCS, pages 334–346, 2009.
30. Kaoru Kurosawa, Ryo Nojima, and Le Trieu Phong. Efficiency-improved fully simulatable adaptive OT under the DDH assumption. In *SCN '10*, volume 6280 of LNCS, pages 172–181, 2010.

31. Fabien Laguillaumie, Pascal Paillier, and Damien Vergnau. Universally convertible directed signature. In *ASIACRYPT '05*, volume 3788 of LNCS, pages 682–701, 2005.
32. Yehuda Lindell. Efficient fully-simulatable oblivious transfer. In *CT-RSA '08*, volume 4964 of LNCS, pages 52–70, 2008.
33. Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *CRYPTO '99*, volume 1666 of LNCS, pages 573–590, 1999.
34. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO '08*, volume 5157, pages 554–571, 2008.
35. Michael Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.
36. Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally composable adaptive priced oblivious transfer. In *Pairing 2009*, volume 5671 of LNCS, pages 231–247, 2009.
37. Claus-Peter Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.
38. Hung-Min Sun, Yalin Chen, and Jue-Sam Chou. An efficient secure oblivious transfer, 2009. Cryptology ePrint Archive: Report 2009/521.
39. S. Wiesner. Conjugate coding. *SIGACT News*, 15:7888, 1983.
40. Andrew Yao. How to generate and exchange secrets. In *FOCS*, pages 162–167, 1986.

A Proof Sketches of Sender and Receiver Security

Complete proofs of sender and receiver security appear in the full version [23].

A.1 Proof Sketch of Sender Security (Theorem 2)

Proof. We will begin with $\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$, then modify the distribution via a series of hybrid games until we arrive at a distribution identical to that of $\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}$. Let us define these hybrids as follows:

Game 0. The real-world experiment conducted between S and $\hat{\mathsf{R}}$ ($\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$).
Game 1. This game modifies **Game 0** as follows: (1) each of S's ZKPoK executions is replaced with a *simulated* proof of the same statement,[8] and (2) the knowledge extractor E_2 is used to obtain the values $(\sigma_i, x, y, z, \bar{c}_4, \bar{c}_5, \bar{c}_6, \bar{c}_7)$ from each of $\hat{\mathsf{R}}$'s transfer queries. Whenever the extractor fails, S terminates the experiment and outputs the distinguished symbol error.
Game 2. This game modifies **Game 1** such that, whenever the extracted value $\sigma_i \in [1, N]$, S's response R is computed using the following approach: parse $C_{\sigma_i} = (c_1, \ldots, c_7)$ and set $R = \frac{c_3 e(g_1, g_2)^x}{M_{\sigma_i}}$. When $\sigma_i \notin [1, N]$, the response is computed using the normal protocol.

[8] This includes at most $k+1$ PoK executions, including the initial $ZKPoK\{(a) : g_1 = g^a\}$ and each subsequent proof $ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.

Game 3. This game modifies **Game 2** by replacing the input to S_I with a dummy vector of random messages $\bar{M}_1, \ldots, \bar{M}_N \in \mathbb{G}_T$. However when S computes a response value using the technique of **Game 2**, the response is based on the original message vector $M_1, \ldots, M_N$. We claim that the distribution of this game is equivalent to that of $\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}$.

Let us now consider the following Lemmas. For notational convenience, define:

$$\mathbf{Adv}\,[\,\mathbf{Game\ i}\,] = \mathbf{Pr}\,[\,D(\mathbf{Game\ i}) = 1\,] - \mathbf{Pr}\,[\,D(\mathbf{Game\ 0}) = 1\,].$$

Lemma 1. *If all p.p.t. algorithms D distinguish a simulated ZKPoK with advantage at most ϵ_{ZK} and the extractor E_2 fails with probability at most ϵ_{Ext}, then* $\mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] \leq (k+1)\cdot\epsilon_{ZK} + k\cdot\epsilon_{Ext}$.

Lemma 2. *If no p.p.t. algorithm has advantage $> \epsilon$ in solving the 3DDH problem, then*

$$\mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] \leq \frac{Np}{p-1}\cdot\epsilon.$$

Lemma 3. *If no p.p.t adversary has advantage $> \epsilon$ at solving the 3DDH problem, then*

$$\mathbf{Adv}\,[\,\mathbf{Game\ 3}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] \leq N\cdot\epsilon.$$

Proof of the above lemmas is in [23]. By summing over hybrids **Game 0** to **Game 3**, we obtain $\mathbf{Adv}\,[\,\mathbf{Game\ 3}\,] \leq (k+1)\epsilon_{ZK} + k\epsilon_{Ext} + N\epsilon(1+\frac{p}{p-1})$. For the Schnorr proofs we use, $\epsilon_{ZK} = 0$. This concludes the proof of Sender security.

A.2 Proof Sketch of Receiver Security (Theorem 3)

Proof. We again arrive at the ideal-world sender via a series of hybrid games:

Game 0. The real-world experiment conducted between $\hat{\mathsf{S}}$ and R ($\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}$).
Game 1. A modification of **Game 0** in which R applies the knowledge extractor E_1 to $\hat{\mathsf{S}}$'s proof ZKPoK$\{a : g_1 = g^a\}$. If this extraction fails, R aborts and outputs $\perp$. Further, for transfers $i = 1$ through k, R uses the knowledge extractor E_3 on $\hat{\mathsf{S}}$'s proof $ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$ to extract the values a, aborting if the extractor fails (or returns inconsistent values).
Game 2. For transfer $i = 1$ to k, modify R's request such that $\sigma_i = 1$. The distribution of this game is identical to that of $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}$.

Lemma 4. *If the extractor E_1 and E_3 fail with probability at most ϵ_{Ext}, then* $\mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] \leq (k+1)\epsilon_{Ext}$.

Lemma 5. *If the Receiver's WIPoK is distinguishable with maximum advantage ϵ_{WI}, then*

$$\mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] =\leq k\cdot\epsilon_{WI}.$$

Proof of the above lemmas is in [23]. Summing the differences, we have

$$\mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 0}\,] = (k+1)\epsilon_{Ext} + k\epsilon_{WI}.$$

For the Schnorr proofs we use, $\epsilon_{WI} = 0$. This concludes the proof of Receiver security.