Efficient, Adaptively Secure, and Composable Oblivious Transfer with a Single, Global CRS

Seung Geol Choi[1⋆], Jonathan Katz[2⋆⋆],
Hoeteck Wee[3⋆⋆⋆], and Hong-Sheng Zhou[2†]

[1] Columbia University
sgchoi@cs.columbia.edu
[2] University of Maryland
{jkatz, hszhou}@cs.umd.edu
[3] George Washington University
hoeteck@alum.mit.edu

Abstract. We present a general framework for efficient, universally composable oblivious transfer (OT) protocols in which a *single*, global, common reference string (CRS) can be used for multiple invocations of oblivious transfer by arbitrary pairs of parties. In addition:

– Our framework is round-efficient. E.g., under the DLIN or SXDH assumptions we achieve round-optimal protocols with static security, or 3-round protocols with adaptive security (assuming erasure).
– Our resulting protocols are more efficient than any known previously, and in particular yield protocols for string OT using $O(1)$ exponentiations and communicating $O(1)$ group elements.

Our result improves on that of Peikert et al. (Crypto 2008), which uses a CRS whose length depends on the number of parties in the network and achieves only static security. Compared to Garay et al. (Crypto 2009), we achieve adaptive security with better round complexity and efficiency.

1 Introduction

In this work we study the construction of efficient protocols for universally composable (UC) [5] oblivious transfer (OT). Our work is motivated by the fact that, although UC *commitments* are complete for UC multiparty

[⋆] This work was done in part at the University of Maryland, and was supported in part by the Intelligence Advanced Research Project Activity (IARPA) via DoI/NBC contract #D11PC20194. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation herein. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

[⋆⋆] This work is supported in part by DARPA and by NSF awards #0447075, #1111599, and #1223623.

[⋆⋆⋆] Supported by NSF CAREER Award CNS-1237429.

[†] Supported by an NSF CI postdoctoral fellowship.

computation [9], the most efficient multiparty computation protocols (e.g., [29, 28]) rely on universally composable OT as a building block. Relative to UC commitments (see [27, 16] and references therein), however, universally composable OT has received less attention.

There is a long series of work on efficient OT protocols in the stand-alone setting (e.g., [30, 1, 21, 25]). Lindell [26] (also [23, Appendix A]) gave a generic transformation from static security to adaptive security (assuming erasure) that applied in the semi-honest setting and the stand-alone malicious setting, but not in the UC setting.

Constructions of UC oblivious transfer from general assumptions were given in [9]; these constructions are relatively inefficient. Garay, MacKenzie, and Yang [17] constructed a constant-round protocol for *committed* OT under the DDH and strong RSA assumptions. Their protocol yields *bit* OT rather than *string* OT, so results in protocols for string OT with complexity linear in the length of the sender's inputs. Jarecki and Shmatikov show a four-round protocol for committed string OT under the decisional composite residuosity (DCR) assumption [24]. A round-optimal OT protocol appears in [22].

The most efficient known protocol for UC oblivious transfer is that of Peikert et al. [33]. Their work, however, has several disadvantages. First, it requires an independent common reference string[4] (CRS) for *every party* in the network or, equivalently, a single CRS of length linear in the number of parties. (Any pair of parties can then run the protocol of Peikert et al. using the CRS of the receiver.) Their protocols also only achieve security against a *static* adversary who decides which parties to corrupt before the protocol begins (and even before the CRS is chosen). They do not handle an *adaptive* adversary who may choose which parties to corrupt during the course of the protocol execution.

Garay et al. [18] constructed efficient UC oblivious-transfer protocols that address both the above-mentioned drawbacks. In their constructions, the parties run a coin-tossing protocol whose outcome is then used as a common random string for an OT protocol. This approach is not entirely satisfactory. First, it increases the overall computation, communication, and round complexity; second, it can (in general) only be instantiated with OT protocols that work in the common *random* string model rather than the more general common *reference* string model. Choi et al. [11, 10] showed other approaches for obtaining adaptively secure, constant-round UC oblivious transfer.

1.1 Our Results

Here, we present a new framework for constructing UC oblivious-transfer protocols that require only a *single*, global CRS. We aim for efficient protocols having low round complexity, and incurring only *constant* computation and communication even when the sender's inputs are long strings. We are also interested in achieving *adaptive* security, under the assumption that parties erase

[4] *Some* form of setup is known to be necessary for universally composable OT [7, 8].

portions of their local state that are no longer needed. (Note, however, that the works of [11, 18, 10] do not make this assumption.)

Our framework is fairly general and can be instantiated from several assumptions. Specifically:

- We obtain efficient, *round-optimal* OT protocols with static security under the decisional linear (DLIN) [3] or symmetric external Diffie-Hellman (SXDH) assumptions [34, 3]. These protocols can be modified to achieve adaptive security (assuming erasure) with one additional round and a slight increase in communication and computation.
- We obtain efficient, four-round OT protocols under the decisional Diffie-Hellman (DDH) or DCR [31] assumptions. Our basic constructions achieve static security, and we present variants that are secure against adaptive corruptions (assuming erasure) *without* any additional rounds, but with a slight increase in communication and computation.

We compare our constructions with previous work in Table 1[5]

Overview of our constructions. The starting point of our approach is the Halevi-Kalai construction [21] of 2-round OT based on smooth projective hashing. Their construction only achieves indistinguishability-based security (and not even stand-alone simulation-based security) against a malicious receiver. We show how to overcome this with the following modifications:

1. We require the receiver to commit to its input using CCA-secure encryption.
2. The receiver proves in zero knowledge that it is behaving consistently in the underlying OT protocol (with respect to the input it committed to).

A similar high-level approach was taken in [22], but using generic simulation-sound non-interactive zero knowledge [15]. Here, following recent constructions of efficient UC commitments [27, 16], we rely instead on efficient zero-knowledge protocols that admit straight-line simulation in the CRS model. In particular, for our two-round OT protocols we instantiate the underlying zero-knowledge proofs using Groth-Sahai proofs [20], as in [16]. For our four-round OT protocols, we rely on Damgård's three-round zero-knowledge proof system [14].

Achieving adaptive security. To achieve adaptive security, we first modify our protocols so the final message is sent over an adaptively secure channel (cf. functionality $\mathcal{F}_{\text{SMT}}$ in [5]). The latter can be realized at low cost if erasure is assumed [2]. With this modification, security against adaptive corruption of the sender is achieved automatically by simply having the sender erase its local state at appropriate times. In our two-round protocols, security against adaptive corruption of the receiver is similarly achieved. For our 4-round protocols, we use techniques similar to those in [27, 16]. Unlike this prior work, however, we do

[5] The numbers for the adaptively secure protocol of [33]+[18]+[27] in Table 1 are based on a preliminary version of [27], and could change once the author publishes the fix to a bug in the protocol.

Reference	Assumption	Rounds	Communication complexity	CRS size
[33]	DDH	2	6	n
[33]+[18]+[16]	DLIN	4	78	12
Protocol 1*	DLIN	2	54	12
[33]+[18]+[27]	DDH	6	38	7
Protocol 2	DDH	4	32	6
[24]	DCR	4	$35\ (\mathbb{Z}_{N^2}) + 16\ (\mathbb{Z}_N)$	10
Protocol 2	DCR	4	$18\ (\mathbb{Z}_{N^2}) + 7\ (\mathbb{Z}_N)$	12

Protocols with static security.

Reference	Assumption	Rounds	Communication complexity	CRS size
[33]+[18]+[16]	DLIN	4	83	12
Protocol 1*	DLIN	3	59	12
[33]+[18]+[27]	DDH	8	51	7
Protocol 2*	DDH	4	35	6
Protocol 2*	DCR	4	$21\ (\mathbb{Z}_{N^2}) + 7\ (\mathbb{Z}_N)$	12

Protocols with adaptive security (assuming erasure).

Table 1. Efficient universally composable protocols for string OT. The number of parties is n. Communication complexity and CRS size are measured in terms of the number of group elements, with other values ignored. The numbers for [24] include the cost of the pre-processing stage.

not introduce any additional overhead in communication or round complexity. (We incur a modest increase in computational cost.)

Organization. We review some preliminaries in Section 2. Our framework for 2-round OT with static security (resp., 3-round OT with adaptive security) is described in Section 3 Our framework for 4-round OT is given in Section 4. Due to space limitations, further details, proofs, and discussions about concrete instantiations have been deferred to the full version.

2 Preliminaries

We let λ be the security parameter. We let $\mathcal{F}_{\mathsf{MOT}}$ be the multi-session OT functionality [5], and $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{P},\mathcal{D}}$ be the CRS functionality [6].

We use the standard notion of chosen-ciphertext security for labeled public-key encryption [4].

$\mathcal{HF} = \{h_k : \{0,1\}^* \to \{0,1\}^{\ell(\lambda)}\}_{k \in \{0,1\}^\lambda}$ is a family of collision-resistant hash functions if for any non-uniform PPT algorithm $\mathcal{A}$, it holds that

$$\Pr[k \leftarrow \{0,1\}^\lambda : \mathcal{A}(k) = (x_1, x_2) \text{ s.t. } x_1 \neq x_2 \text{ and } h_k(x_1) = h_k(x_2)] = \mathsf{negl}(\lambda).$$

2.1 Smooth Projective Hash Proof Systems

We recall the notion of a hard subset membership problem and smooth projective hashing defined by Cramer and Shoup [13], following the notation of [21]. A hash family $\mathcal{H}$ consists of the following PPT algorithms:

- The *parameter-generator* $\mathsf{HashPG}(1^\lambda) \rightarrow \textsc{pp}$. We assume that the security parameter λ can be inferred from $\textsc{pp}$. Let $\lambda(\textsc{pp})$ denote the security parameter corresponding to $\textsc{pp}$.
- A pair of disjoint sets $\Lambda_{\textsc{yes}}$ and $\Lambda_{\textsc{no}}$ are associated to $\textsc{pp}$ corresponding to $\textsc{yes}$ and $\textsc{no}$ instances respectively. There exists a $\textsc{yes}$ *instance-sampler* $\mathsf{SampYes}(\textsc{pp}) \rightarrow (x, w)$ where x is uniformly distributed over $\Lambda_{\textsc{yes}}$ and w is the corresponding witness. There also exists a $\textsc{no}$ *instance-sampler* $\mathsf{SampNo}(\textsc{pp}) \rightarrow x'$ where x' is uniformly distributed over $\Lambda_{\textsc{no}}$.
- The *hash-key generator* $\mathsf{HashKG}(\textsc{pp}) \rightarrow (\textsc{hk}, \textsc{pk})$. Here $\textsc{hk}$ is the primary hashing key and $\textsc{pk}$ is a projective key.
- The *primary hash algorithm* $\mathsf{Hash}(\textsc{hk}, x) \rightarrow y$ for all $x \in \Lambda_{\textsc{yes}} \cup \Lambda_{\textsc{no}}$.
- The *projection hash algorithm* $\mathsf{pHash}(\textsc{pk}, x, w) \rightarrow y$ for all $(x, w) \leftarrow \mathsf{SampYes}(\textsc{pp})$.

We require that for all $\textsc{pp} \in \mathsf{support}(\mathsf{HashPG})$, every $(\textsc{hk}, \textsc{pk}) \leftarrow \mathsf{HashKG}(\textsc{pp})$, and every $(x, w) \leftarrow \mathsf{SampYes}(\textsc{pp})$, we have $\mathsf{pHash}(\textsc{pk}, x, w) = \mathsf{Hash}(\textsc{hk}, x)$.

Definition 1. $\mathcal{H} = (\mathsf{HashPG}, \mathsf{SampYes}, \mathsf{SampNo}, \mathsf{HashKG}, \mathsf{Hash}, \mathsf{pHash})$ *is a* smooth projective hash family *if*

Smoothness: *Let* $(\textsc{hk}, \textsc{pk}) \leftarrow \mathsf{HashKG}(\textsc{pp})$*. For all* $x \in \Lambda_{\textsc{no}}$*, the distribution of* $\mathsf{Hash}(\textsc{hk}, x)$ *given* $\textsc{pk}$ *is statistically close to uniform. That is, the statistical difference between the following two distributions is negligible in* $\lambda(\textsc{pp})$*.*

$$\{y \leftarrow \mathsf{Hash}(\textsc{hk}, x) :\ (\textsc{pk}, y, x)\} \overset{s}{\equiv} \{y \leftarrow \Gamma :\ (\textsc{pk}, y, x)\}$$

where Γ *denotes the set of possible hash values with parameter* $\textsc{pp}$*.*

Definition 2. *A smooth projective hash family* $\mathcal{H} = (\mathsf{HashPG}, \mathsf{SampYes}, \mathsf{SampNo}, \mathsf{HashKG}, \mathsf{Hash}, \mathsf{pHash})$ *is said to have a* hard subset membership *property if the following two ensembles are computationally indistinguishable:*

- $\left\{\textsc{pp} \leftarrow \mathsf{HashPG}(1^\lambda);\ (x, w) \leftarrow \mathsf{SampYes}(\textsc{pp}) :\ (\textsc{pp}, x)\right\}_{\lambda \in \mathbb{N}}$
- $\left\{\textsc{pp} \leftarrow \mathsf{HashPG}(1^\lambda);\ x \leftarrow \mathsf{SampNo}(\textsc{pp}) :\ (\textsc{pp}, x)\right\}_{\lambda \in \mathbb{N}}$.

2.2 Dual-Mode NIZK

Groth introduced non-interactive zero-knowledge (NIZK) proofs [19] that we call *dual-mode*. In such a proof system, a common reference string crs is generated in either a *soundness* mode or a *zero-knowledge* (ZK) mode; given crs, it is infeasible to determine the mode in which it was generated. When crs is generated in the soundness mode, the proof system is statistically sound. On the other hand, when crs is generated in the ZK mode, the simulation is perfect. Groth and Sahai [20] provide efficient dual-mode NIZK proofs for various equations in bilinear groups.

Definition 3. *A* non-interactive proof system *for a language $L \in \mathcal{NP}$ consists of three algorithms $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ where $\mathcal{K}$ is a CRS generation algorithm, $\mathcal{P}$ and $\mathcal{V}$ are a prover and a verifier algorithm respectively. The system is required to satisfy the following properties:*

Completeness: *For any λ, any $x \in L$, and any witness w for x, it holds that*

$$\Pr[\mathsf{crs} \leftarrow \mathcal{K}(1^\lambda);\ \pi \leftarrow \mathcal{P}(1^\lambda, \mathsf{crs}, x, w) :\ \mathcal{V}(1^\lambda, \mathsf{crs}, x, \pi) = 1] = 1.$$

Adaptive soundness: *For any λ and any adversary $\mathcal{A}$, it holds that*

$$\Pr[\mathsf{crs} \leftarrow \mathcal{K}(1^\lambda);\ (x, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}) :\ \mathcal{V}(1^\lambda, \mathsf{crs}, x, \pi) = 1\ \wedge x \notin L] = \mathsf{negl}(\lambda).$$

Definition 4. *A non-interactive proof system $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ for a language $L \in \mathcal{NP}$ is said to be* dual-mode NIZK *if there is a pair of efficient algorithms $(\mathcal{S}_1, \mathcal{S}_2)$ such that for any $\lambda \in \mathbb{N}$ and for all non-uniform polynomial time adversary $\mathcal{A}$, it holds the following:*

Indistinguishability of modes:

$$\left|\Pr[\mathsf{crs} \leftarrow \mathcal{K}(1^\lambda) :\ \mathcal{A}(1^\lambda, \mathsf{crs}) = 1] - \Pr[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\lambda) :\ \mathcal{A}(1^\lambda, \mathsf{crs}) = 1]\right| = \mathsf{negl}(\lambda).$$

Perfect simulation in ZK mode: *The following two probabilities are equal.*
- $\Pr[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\lambda);\ (x, w) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}, \tau);\ \pi \leftarrow \mathcal{P}(1^\lambda, \mathsf{crs}, x, w) : \mathcal{A}(\pi) = 1]$
- $\Pr[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\lambda);\ (x, w) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}, \tau);\ \pi \leftarrow \mathcal{S}_2(\tau, x) : \mathcal{A}(\pi) = 1]$

Here, $\mathcal{A}$ has to generate a pair (x, w) with w a witness for x.

2.3 Σ-Protocols

A Σ-protocol is a 3-round honest-verifier zero-knowledge protocol. We denote by (a, e, z) the messages exchanged between the prover $\mathcal{P}_\Sigma$ and the verifier $\mathcal{V}_\Sigma$. We say a transcript (a, e, z) is an *accepting transcript for x* if $\mathcal{V}_\Sigma$ would accept based on the values (x, a, e, z). We use the standard definitions of special soundness and special honest-verifier zero knowledge.

2.4 Equivocal Commitments

We define an equivocal commitment scheme as follows:

Definition 5. *Let $(\mathcal{K}_{com}, \mathsf{Com})$ be a non-interactive commitment scheme with CRS where $\mathcal{K}_{com}$ is a CRS generation algorithm, and* Com *is a commitment algorithm. The scheme is said to be* equivocal *if there exists a tuple of PPT algorithm $(\mathcal{S}_{com1}, \mathcal{S}_{com2}, \mathcal{S}_{com3})$ that satisfies the following properties:*

Computational binding: *For any non-uniform polynomial time adversary $\mathcal{A}$ the following is negligible in λ:*

$$\Pr\left[\begin{array}{r}\mathsf{crs} \leftarrow \mathcal{K}_{com}(1^\lambda);\ (m, m', r, r') \leftarrow \mathcal{A}(\mathsf{crs}) :\\ m \neq m' \bigwedge \mathsf{Com}_{\mathsf{crs}}(m; r) = \mathsf{Com}_{\mathsf{crs}}(m'; r')\end{array}\right]$$

Indistinguishability of modes:

$$\left\{\mathsf{crs}\leftarrow\mathcal{K}_{com}(1^\lambda):\ \mathsf{crs}\right\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \left\{(\mathsf{crs},t)\leftarrow\mathcal{S}_{com1}(1^\lambda):\ \mathsf{crs}\right\}_{\lambda\in\mathbb{N}}$$

Equivocality: *For any* $\lambda\in\mathbb{N}$*, any* $(\mathsf{crs},t)\in\mathsf{support}(\mathcal{S}_{com1}(1^\lambda))$*, and any adversary* $\mathcal{A}$*, the following distributions are identical.*

- $\left\{m\leftarrow\mathcal{A}(\mathsf{crs});\ r\leftarrow\mathcal{R};\ c=\mathsf{Com}_{\mathsf{crs}}(m;r):\ (m,r,c)\right\}$
- $\left\{m\leftarrow\mathcal{A}(\mathsf{crs});\ (c,s)\leftarrow\mathcal{S}_{com2}(t);\ r\leftarrow\mathcal{S}_{com3}(s,m):\ (m,r,c)\right\}$

3 A Generic Framework for Two-Round OT

In this section we describe **Protocol 1***, an adaptively secure, 2-round protocol. Let $(\mathcal{K},\mathcal{P},\mathcal{V})$ be a dual-mode NIZK proof system, $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be a CCA-secure labeled public-key encryption scheme, and $\mathcal{H}=(\mathsf{HashPG},\mathsf{SampYes},$ $\mathsf{SampNo},\ \mathsf{HashKG},\ \mathsf{Hash},\mathsf{pHash})$ be a smooth hash proof system with a hard subset membership property. We assume for simplicity that $\{0,1\}^\ell$ is the range of the hash functions in $\mathcal{H}$; known constructions can be modified to achieve this property. Based on these components, we construct an OT protocol between a sender P_i and a receiver P_j in the CRS model; refer also to Figure 1.

Common reference string: Compute $\textsc{pp}\leftarrow\mathsf{HashPG}(1^\lambda)$, $(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda)$, and $\mathsf{crs}_{nizk}\leftarrow\mathcal{K}(1^\lambda)$. The common reference string is $\mathsf{crs}_{ot}=(\textsc{pp},pk,\mathsf{crs}_{nizk})$.

Oblivious transfer: The protocol starts by having the receiver, holding selection bit b, send two instances (x_0,x_1) for the hash proof system $\mathcal{H}$ with x_{1-b} a NO-instance; the receiver sends $\mathsf{Enc}_{pk}(b)$ and a NIZK proof that x_{1-b} is a NO-instance as well. In the second round, for $\sigma\in\{0,1\}$ the sender generates primary and projection hash keys $(\textsc{hk}_\sigma,\textsc{pk}_\sigma)$ and sends $(\textsc{pk}_\sigma,\mathsf{Hash}(\textsc{hk}_\sigma,x_\sigma)\oplus m_\sigma)$ to the receiver. The receiver recovers m_b in the standard way. In more detail:

- On input a selection bit b, the receiver P_j proceeds as follows:
  1. Compute $(x_b,w)\leftarrow\mathsf{SampYes}(\textsc{pp})$ and $x_{1-b}=\mathsf{SampNo}(\textsc{pp};\gamma)$ for uniform γ. Compute $\Phi=\mathsf{Enc}^L_{pk}(b;\xi)$ with uniformly random ξ, where $L=(\mathsf{sid},\mathsf{ssid},P_i,P_j)$. Generate an NIZK proof π that there exist (b,γ,ξ) such that $x_{1-b}=\mathsf{SampNo}(\textsc{pp};\gamma)$ and $\Phi=\mathsf{Enc}^L_{pk}(b;\xi)$.
  2. Send $\langle x_0,x_1,\Phi,\pi\rangle$.
- On input $m_0,m_1\in\{0,1\}^\ell$, and after receiving the first-round message $\langle x_0,x_1,\Phi,\pi\rangle$ from the receiver, the sender P_i proceeds as follows:
  1. If the proof π does not verify, abort.
  2. For $\sigma\in\{0,1\}$ compute $(\textsc{hk}_\sigma,\textsc{pk}_\sigma)\leftarrow\mathsf{HashKG}(\textsc{pp})$ and $Z_\sigma=m_\sigma\oplus\mathsf{Hash}(\textsc{hk}_\sigma,x_\sigma)$.
  3. Send $\langle\textsc{pk}_0,Z_0,\textsc{pk}_1,Z_1\rangle$ to P_j.
- Upon receiving the second-round message $\langle\textsc{pk}_0,Z_0,\textsc{pk}_1,Z_1\rangle$, the receiver P_j computes the output $m_b=Z_b\oplus\mathsf{pHash}(\textsc{pk}_b,x_b,w)$.

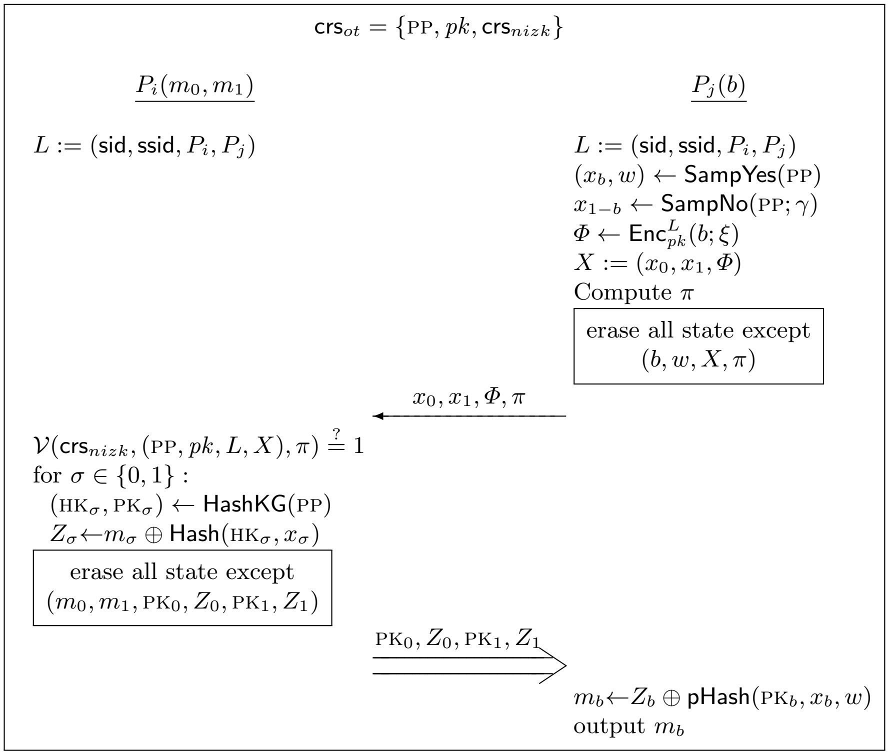
$\mathsf{crs}_{ot} = \{\textsc{pp}, pk, \mathsf{crs}_{nizk}\}$
$P_i(m_0, m_1)$
$P_j(b)$
$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
$(x_b, w) \leftarrow \mathsf{SampYes}(\textsc{pp})$
$x_{1-b} \leftarrow \mathsf{SampNo}(\textsc{pp}; \gamma)$
$\Phi \leftarrow \mathsf{Enc}^L_{pk}(b; \xi)$
$X := (x_0, x_1, \Phi)$
Compute π
erase all state except (b, w, X, π)
x_0, x_1, Φ, π
$\mathcal{V}(\mathsf{crs}_{nizk}, (\textsc{pp}, pk, L, X), \pi) \stackrel{?}{=} 1$
for $\sigma \in \{0,1\}$:
$(\textsc{hk}_\sigma, \textsc{pk}_\sigma) \leftarrow \mathsf{HashKG}(\textsc{pp})$
$Z_\sigma \leftarrow m_\sigma \oplus \mathsf{Hash}(\textsc{hk}_\sigma, x_\sigma)$
erase all state except $(m_0, m_1, \textsc{pk}_0, Z_0, \textsc{pk}_1, Z_1)$
$\textsc{pk}_0, Z_0, \textsc{pk}_1, Z_1$
$m_b \leftarrow Z_b \oplus \mathsf{pHash}(\textsc{pk}_b, x_b, w)$
output m_b

Fig. 1. An OT protocol in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid model (**Protocol 1***). For adaptive security, the second-round message is sent over an adaptively secure channel.

Informally, security against a malicious sender holds because the sender cannot guess the receiver's selection bit due to the hard subset membership property. On the other hand, a malicious receiver gets no information about m_{1-b} if x_{1-b} is a NO-instance, and this property is enforced by the NIZK proof.

Theorem 1. *Say* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a CCA-secure labeled public-key encryption scheme,* $(\mathsf{HashPG}, \mathsf{SampYes}, \mathsf{SampNo}, \mathsf{HashKG}, \mathsf{Hash}, \mathsf{pHash})$ *is a smooth projective hash proof system with hard subset membership property, and* $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ *is a dual-mode NIZK proof system. Then the protocol described above securely realizes* $\mathcal{F}_{\mathsf{MOT}}$ *in the* $\mathcal{F}_{\mathsf{CRS}}$*-hybrid model, for static corruptions. If the second round message is sent over an adaptively secure channel, the protocol securely realizes* $\mathcal{F}_{\mathsf{MOT}}$ *in the* $\mathcal{F}_{\mathsf{CRS}}$*-hybrid model, for adaptive corruptions (assuming erasure).*

In the full version of this work, we discuss concrete instantiations of this framework based on the DLIN and SXDH assumptions.

$\mathsf{crs}_{ot} = \{\textsc{pp}, pk, \mathsf{crs}_{com}\}$

$P_i(m_0, m_1)$		$P_j(b)$
$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$		$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ $(x_b, w) \leftarrow \mathsf{SampYes}(\textsc{pp})$ $x_{1-b} \leftarrow \mathsf{SampNo}(\textsc{pp}; \gamma)$ $\Phi \leftarrow \mathsf{Enc}^L_{pk}(b; \xi); X := (x_0, x_1, \Phi)$ $a \leftarrow \mathcal{P}_\Sigma((\textsc{pp}, pk, L, X), (b, \gamma, \xi))$ $c \leftarrow \mathsf{Com}_{\mathsf{crs}_{com}}(a; r)$
	$\xleftarrow{x_0, x_1, \Phi, c}$	
$e \leftarrow \{0,1\}^\lambda$		
	$\xrightarrow{e}$	$z \leftarrow \mathcal{P}_\Sigma((\textsc{pp}, pk, L, X), (b, \gamma, \xi), e)$
$\mathcal{V}_\Sigma((\textsc{pp}, pk, L, X), a, e, z) \overset{?}{=} 1$	$\xleftarrow{(a, r), z}$	
$\mathsf{Com}_{\mathsf{crs}_{com}}(a; r) \overset{?}{=} c$ for $\sigma \in \{0,1\}$: $(\textsc{hk}_\sigma, \textsc{pk}_\sigma) \leftarrow \mathsf{HashKG}(\textsc{pp})$		
$Z_\sigma \leftarrow m_\sigma \oplus \mathsf{Hash}(\textsc{hk}_\sigma, x_\sigma)$	$\xrightarrow{\textsc{pk}_0, Z_0, \textsc{pk}_1, Z_1}$	$m_b \leftarrow Z_b \oplus \mathsf{pHash}(\textsc{pk}_b, x_b, w)$ output m_b

Fig. 2. A statically secure OT protocol in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid model (**Protocol 2**).

4 A Generic Framework for Four-Round OT

In this section, we describe a generic framework for constructing four-round OT protocols. We begin by looking at the case of static security, and then show how the ideas can be extended to achieve security against adaptive adversaries.

4.1 Static Security (Protocol 2)

The main idea is to adapt our previous two-round framework by replacing the dual-mode NIZK proof with an interactive equivalent. In particular, the general structure of the protocol is as follows: the protocol starts by having the receiver send two instances (x_0, x_1) for hash proof system where x_{1-b} being a NO-instance; also, in protection against a malicious behavior, $\mathsf{Enc}_{pk}(b)$ and a Sigma protocol (augmented with an equivocal commitment) are attached. Then, the sender generates primary and projective hash keys $(\textsc{hk}_\sigma, \textsc{pk}_\sigma)$ for each instance x_σ and sends $(\textsc{pk}_\sigma, \mathsf{Hash}(\textsc{hk}_\sigma, x_\sigma) \oplus m_\sigma)$ to the receiver. The security can be shown similarly to the two-round OT case.

Here, instead of replicating all the details, we only describe how to combine a Sigma protocol with an equivocal commitment scheme in order to replace the NIZK part. The idea is having the prover commit to the first round message of the Sigma protocol, and reveal it in the third round. Refer to Figure 2 for the overall pictorial description of the protocol.

CRS. Compute $\textsc{pp} \leftarrow \mathsf{HashPG}(1^\lambda)$, $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, and $\mathsf{crs}_{com} \leftarrow \mathcal{K}_{com}(1^\lambda)$. The common reference string is $\mathsf{crs}_{ot} = (\textsc{pp}, pk, \mathsf{crs}_{com})$.

Replacing NIZK. Recall in the two-round OT case, the receiver generates a NIZK π to prove that (x_0, x_1, Φ) is valid message, i.e., Φ is an encryption of $b \in \{0,1\}$ for some b and x_{1-b} is NO-instance. In this protocol, the receiver proves it by running a Sigma protocol $(\mathcal{P}_\Sigma, \mathcal{V}_\Sigma)$, along with an equivocal commitment scheme $(\mathcal{K}_{com}, \mathsf{Com})$, with respect to the following language:

$$\mathcal{L}^* = \left\{ \begin{array}{l} (\textsc{pp}, pk, L, x_0, x_1, \Phi) : \\ \qquad \exists (b, \gamma, \xi) \text{ s.t. } x_{1-b} = \mathsf{SampNo}(\textsc{pp}; \gamma), \Phi = \mathsf{Enc}^L_{pk}(b; \xi) \end{array} \right\},$$

where $L = (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$.

1. The receiver runs $a \leftarrow \mathcal{P}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), (b, \gamma, \xi))$, and computes $c = \mathsf{Com}_{\mathsf{crs}_{com}}(a; r)$ with r chosen uniformly at random. It sends (x_0, x_1, Φ, c).
2. The sender sends the challenge message $e \leftarrow \{0,1\}^\lambda$ of the Sigma protocol.
3. Upon receiving the challenge e, the receiver generates an answer by running
$$z = \mathcal{P}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), (b, \gamma, \xi), e).$$
It sends the sender the answer z along with the opening of the commitment, i.e., $((a, r), z)$.
4. The sender verifies (a, e, z) is an accepting transcript and (a, r) is a valid opening of c:
$$\mathcal{V}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), a, e, z) \stackrel{?}{=} 1, \quad \mathsf{Com}_{\mathsf{crs}_{com}}(a; r) \stackrel{?}{=} c.$$

The security of the protocol can be proved similarly to the two-round case.

Theorem 2. *Say* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a CCA-secure labeled public-key encryption scheme,* $(\mathsf{HashPG}, \mathsf{SampYes}, \mathsf{SampNo}, \mathsf{HashKG}, \mathsf{Hash}, \mathsf{pHash})$ *is a smooth projective hash proof system with hard subset membership property,* $(\mathcal{P}_\Sigma, \mathcal{V}_\Sigma)$ *is a* Σ*-protocol, and* $(\mathcal{K}_{com}, \mathsf{Com})$ *is an equivocal commitment scheme. Then the protocol of Figure 2 securely realizes* $\mathcal{F}_{\mathsf{MOT}}$ *in the* $\mathcal{F}_{\mathsf{CRS}}$*-hybrid model, for static corruptions.*

4.2 Adaptive Security (Protocol 2*)

As with the 2-round framework, the protocol first needs to be changed so that the last round message is sent over a secure channel. This modification (along with erasing the state appropriately), however, is not sufficient to deal with adaptive corruption in the four-round case. For the NIZK, the receiver can generate π and then erase the unnecessary internal state before sending out (x_0, x_1, Φ, π). However, if the statement is composed with the interactive Sigma protocol, some of the internal state cannot be erased until the last move. For example, in the Sigma protocol, the receiver cannot erase the randomness used for generating the NO-instance x_{1-b} until it receives the challenge e, since he has

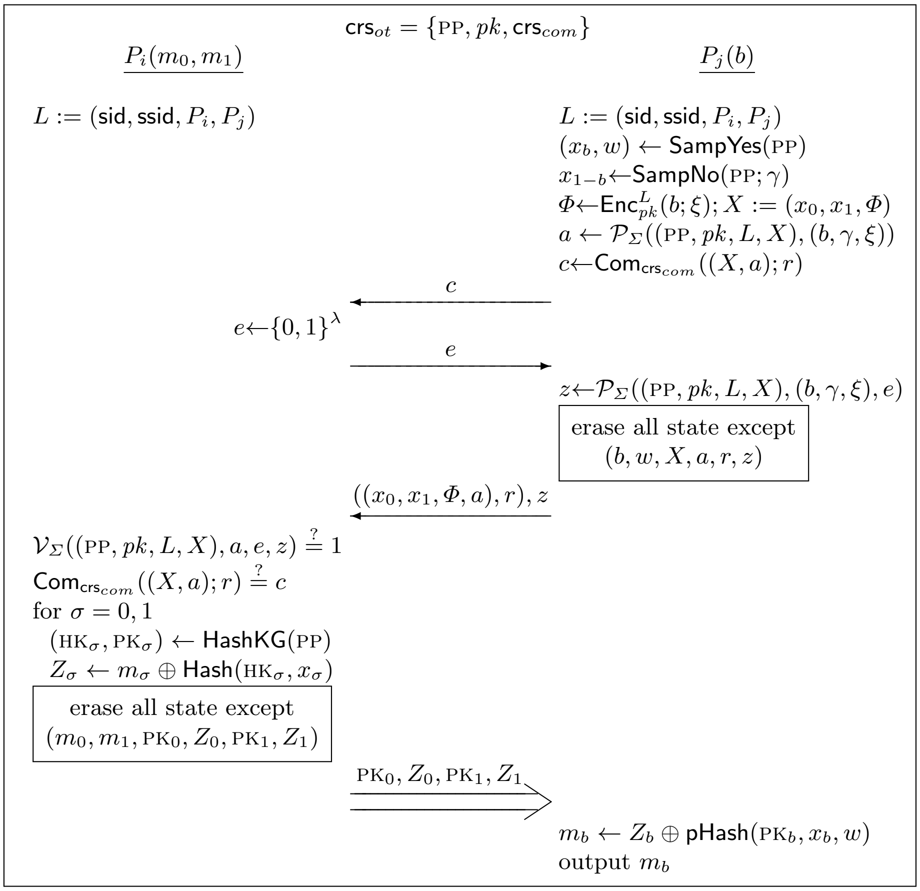
$\mathsf{crs}_{ot} = \{\text{PP}, pk, \mathsf{crs}_{com}\}$
$P_i(m_0, m_1)$
$P_j(b)$
$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
$L := (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
$(x_b, w) \leftarrow \mathsf{SampYes}(\text{PP})$
$x_{1-b} \leftarrow \mathsf{SampNo}(\text{PP}; \gamma)$
$\Phi \leftarrow \mathsf{Enc}^L_{pk}(b; \xi); X := (x_0, x_1, \Phi)$
$a \leftarrow \mathcal{P}_\Sigma((\text{PP}, pk, L, X), (b, \gamma, \xi))$
$c \leftarrow \mathsf{Com}_{\mathsf{crs}_{com}}((X, a); r)$
c
$e \leftarrow \{0,1\}^\lambda$
e
$z \leftarrow \mathcal{P}_\Sigma((\text{PP}, pk, L, X), (b, \gamma, \xi), e)$
erase all state except (b, w, X, a, r, z)
$((x_0, x_1, \Phi, a), r), z$
$\mathcal{V}_\Sigma((\text{PP}, pk, L, X), a, e, z) \overset{?}{=} 1$
$\mathsf{Com}_{\mathsf{crs}_{com}}((X, a); r) \overset{?}{=} c$
for $\sigma = 0, 1$
$(\text{HK}_\sigma, \text{PK}_\sigma) \leftarrow \mathsf{HashKG}(\text{PP})$
$Z_\sigma \leftarrow m_\sigma \oplus \mathsf{Hash}(\text{HK}_\sigma, x_\sigma)$
erase all state except $(m_0, m_1, \text{PK}_0, Z_0, \text{PK}_1, Z_1)$
$\text{PK}_0, Z_0, \text{PK}_1, Z_1$
$m_b \leftarrow Z_b \oplus \mathsf{pHash}(\text{PK}_b, x_b, w)$
output m_b

Fig. 3. An adaptively secure OT protocol in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid model (**Protocol 2***). The final message is sent over an adaptively secure channel.

to use the randomness as part of the witness in order to finish the proof. However, recall that both x_0 and x_1 are YES instances in simulation; when the adversary corrupts the receiver right before sending e, the simulator cannot return a valid randomness for x_{1-b}, and so the simulation breaks down.

Changing the order of messages. As in the commitment scheme [27], we resolve this issue by switching the order of messages. That is, the message to be committed to is not only the first message a of the Sigma protocol but also the statement itself (i.e., (x_0, x_1, Φ)), and they are revealed at the last move of the Sigma protocol. Now, thanks to the equivocality of the commitment scheme, the protocol can achieve adaptive security. Refer to Figure 3 for the overall pictorial description. Here, we only describe the aforementioned modification in more detail. Recall in the statically secure protocol described in Section 4.1, the receiver sends (x_0, x_1, Φ) and the commitment c to the first message a of the Sigma protocol $(\mathcal{P}_\Sigma, \mathcal{V}_\Sigma)$ for the language

$$\mathcal{L}^* = \left\{ \begin{array}{l} (\textsc{pp}, pk, L, x_0, x_1, \Phi) : \\ \quad \exists(b, \gamma, \xi) \text{ s.t. } x_{1-b} = \mathsf{SampNo}(\textsc{pp}; \gamma), \Phi = \mathsf{Enc}^L_{pk}(b; \xi) \end{array} \right\},$$

where $L = (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$. In this protocol, we change the order of messages as follows:

1. The receiver runs $a \leftarrow \mathcal{P}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), (b, \gamma, \xi))$, and then computes $c \leftarrow \mathsf{Com}_{\mathsf{crs}_{com}}((x_0, x_1, \Phi, a); r)$ with r chosen uniformly at random. It sends c.
2. The sender sends the challenge message $e \leftarrow \{0,1\}^\lambda$ of the Sigma protocol.
3. Upon receiving the challenge e, the receiver generates an answer by running
$$z = \mathcal{P}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), (b, \gamma, \xi), e).$$
It sends the sender the answer z along with the opening of the commitment, i.e., $((x_0, x_1, \Phi, a), r, z)$.
4. The sender verifies (a, e, z) is an accepting transcript and $((x_0, x_1, \Phi, a), r)$ is a valid opening of c:
$$\mathcal{V}_\Sigma((\textsc{pp}, pk, L, x_0, x_1, \Phi), a, e, z) \overset{?}{=} 1, \quad \mathsf{Com}_{\mathsf{crs}_{com}}((x_0, x_1, \Phi, a); r) \overset{?}{=} c.$$

Theorem 3. *Under the same assumptions as in Theorem 2, the protocol in Figure 3 securely realizes $\mathcal{F}_{\mathsf{MOT}}$ in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid model, for adaptive corruptions (assuming erasure).*

4.3 Instantiations from the DDH Assumption

We show a CCA-secure labeled public-key encryption scheme, a smooth hash proof system, and an equivocal commitment scheme under the DDH assumption. We then obtain a four-round OT protocol by combining these building blocks.

Decisional Diffie-Hellman assumption. Let $\mathcal{G}_{\rm ddh}$ be a randomized algorithm that takes a security parameter λ and outputs $\mathsf{desc} = (p, \mathbb{G}, g)$ such that $\mathbb{G}$ is the description of group of prime order p, and g is a generator of $\mathbb{G}$.

Definition 6. *The* DDH problem is hard relative to $\mathbb{G}$ *if for all* PPT *algorithms* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr[\mathcal{A}(\mathbb{G}, p, g, g^a, g^b, g^c) = 1] - \Pr[\mathcal{A}(\mathbb{G}, p, g, g^a, g^b, g^{ab}) = 1]\right| \le \mathsf{negl}(\lambda)$$

where in each case the probabilities are taken over the experiment in which the group-generating algorithm outputs $(\mathbb{G}, p, g)$ *and random* $a, b, c \in \mathbb{Z}_p$ *are chosen.*

CCA-secure labeled public-key encryption. Since the DDH assumption holds in $\mathbb{G}_1$, we can use Cramer-Shoup encryption scheme [12]. As in the case for the DLIN assumption, we slightly change the scheme to support labels, that is, we use collision resistant hash functions instead of UOWHF and apply labels to hash functions when performing encryptions and decryptions.

Key generation $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{desc})$: Choose random generators $g_1 \leftarrow \mathbb{G}$ and exponents $\beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2 \leftarrow \mathbb{Z}_p$ and compute $c = g_1^{\beta_1} g^{\beta_2}, d = g_1^{\gamma_1} g^{\gamma_2}, h = g_1^{\delta_1} g^{\delta_2}$. Choose a hash function $H \leftarrow \mathcal{HF}$ where $\mathcal{HF}$ is a family of collision-resistant hash functions. Now set $pk = (g_1, g, c, d, h, H)$ and $sk = (\beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2)$.

Encryption $C \leftarrow \mathsf{Enc}_{pk}^L(m; r)$: Given the message $m \in \mathbb{G}$ under label L, choose $r \leftarrow \mathbb{Z}_p$ and compute $u_1 = g_1^r, u_2 = g^r, e = m \cdot h^r$. Then compute $\alpha = H(u_1, u_2, e, L) \in \mathbb{Z}_p$ and $v = (cd^\alpha)^r$. The ciphertext is $C = (u_1, u_2, e, v)$.

Decryption $\mathsf{Dec}_{sk}^L(C)$: Parse $C = (u_1, u_2, e, v)$ and $sk = (\beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2)$; compute $\alpha \leftarrow H(u_1, u_2, e, L)$ and test if $u_1^{\beta_1 + \alpha\gamma_1} \cdot u_2^{\beta_2 + \alpha\gamma_2} \overset{?}{=} v$. If it does not, output `reject`. Otherwise, output $m = e/(u_1^{\delta_1} u_2^{\delta_2})$.

Smooth projective hashing. We recall the smooth projective hashing based on the DDH assumption [12, 13].

Parameter generation. Choose $g_1, g \leftarrow \mathbb{G}$. Then PP $= (g_1, g, \mathbb{G})$.

Instance sampling. To sample a YES instance, choose $t \leftarrow \mathbb{Z}_p$, and compute $z_1 = g_1^t$, $z_2 = g^t$, and then return $x = (z_1, z_2)$. To sample a NO instance, choose $t \leftarrow \mathbb{Z}_p$, and then $z_1 = g_1^t$, $z_2 = g^{t+1}$, and then return $x = (z_1, z_2)$.

Hash key generation. Choose $\theta_1, \theta_2 \leftarrow \mathbb{Z}_p$ and compute $f = g_1^{\theta_1} g^{\theta_2}$. Return HK $= (\theta_1, \theta_2)$, and PK $= f$.

Primary hashing. Given HK $= (\theta_1, \theta_2)$ and $x = (z_1, z_2)$, return $y = z_1^{\theta_1} z_2^{\theta_2}$.

Projective hashing. Given a projective hash key PK $= f$, an instance $x = (z_1, z_2)$, and its witness $w = t$ such that $z_1 = g_1^t$, $z_2 = g^t$, return $y = f^t$.

Equivocal commitment. We use a variant of the Pedersen commitment scheme [32]. The main difference from the original Pedersen commitment is that collision resilient hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ is used to commit to arbitrary long message very efficiently. In particular, given the CRS $(g, h_1) \in \mathbb{G}^2$, the commitment to a message m is $g^r h_1^{H(m)}$. We note that the binding property is under the DLOG assumption and the collision resilient property of the hash function. When a trapdoor ζ with $h_1 = g^\zeta$ is known, it easy to equivocate a commitment $c = g^s$ into any m by outputting $r = s - \zeta \cdot H(m)$.

By plugging these components into our generic framework for four-round OT, we obtain an OT protocol based on the DDH assumption. It is only left to show the concrete Σ-protocol that is used.

Protocol details. Ignoring the description desc of the group $\mathbb{G}$, the CRS is $\mathsf{crs}_{ot} = (\text{PP}, pk, \mathsf{crs}_{com})$ where $\text{PP} = (g_1, g)$ $\quad pk = (g_1, g, c, d, h, H)$ $\quad \mathsf{crs}_{com} = (h_1, g)$. Therefore, the CRS can be represented with 6 group elements of $\mathbb{G}$ and one hash function index, along with the description of the group $\mathbb{G}$.

Let $x_0 = (z_{01}, z_{02})$, $x_1 = (z_{11}, z_{12})$, and $\Phi = (u_1, u_2, e, v)$ with $\alpha = H(u_1, u_2, e, (\mathsf{sid}, \mathsf{ssid}, P_i, P_j))$. Then, we use a standard Sigma protocol for the following language:

$$\mathcal{L}^* = \left\{ \begin{array}{l} (\mathsf{crs}_{ot}, pk, x_0, x_1, \Phi, \alpha) : \\ \exists (r,t) \text{ s.t. } u_1 = g_1^r, u_2 = g^r, e = h^r, v = (cd^\alpha)^r, z_{11} = g_1^t, z_{12} = g^{t+1} \\ \qquad\qquad \text{or } u_1 = g_1^r, u_2 = g^r, e = gh^r, v = (cd^\alpha)^r, z_{01} = g_1^t, z_{02} = g^{t+1} \end{array} \right\}.$$

1. Suppose that $\Phi = \mathsf{Enc}(g^b)$. Let $\bar{b} = 1 - b$. The prover chooses $R, T \leftarrow \mathbb{Z}_p$, $\eta \leftarrow [0, 2^\lambda)$, and $\rho, \tau \leftarrow \mathbb{Z}_p$. Then, it computes and sends the verifier the following:

$$\begin{array}{lll} U_{1b} = g_1^R, & U_{2b} = g^R, & E_b = h^R, \\ V_b = (cd^\alpha)^R, & Z_{1b} = g_1^T, & Z_{2b} = g^T \\ U_{1\bar{b}} = g_1^\rho / u_1^\eta, & U_{2\bar{b}} = g_1^\rho / u_2^\eta, & E_{\bar{b}} = h^\rho / (e/g^{\bar{b}})^\eta, \\ V_{\bar{b}} = (cd^\alpha)^\rho / v^\eta, & Z_{1\bar{b}} = g_1^\tau / z_{b1}^\eta, & Z_{2\bar{b}} = g^\tau / (z_{b2}/g)^\eta. \end{array}$$

2. The verifier chooses $\epsilon \leftarrow [0, 2^\lambda)$ and sends it to the prover.
3. The prover computes the following:

$$\begin{array}{ll} \epsilon_b = \epsilon - \eta \bmod 2^\lambda & \epsilon_{\bar{b}} = \eta \\ \rho_b = R + r\epsilon_b & \rho_{\bar{b}} = \rho \\ \tau_b = T + t\epsilon_b & \tau_{\bar{b}} = \tau. \end{array}$$

   Then, it sends $(\epsilon_0, \rho_0, \tau_0, \rho_1, \tau_1)$ to the verifier.
4. The verifier computes $\epsilon_1 = \epsilon - \epsilon_0 \bmod 2^\lambda$. It also checks if the following holds for $i \in \{0, 1\}$.

$$\begin{array}{lll} g_1^{\rho_i} = U_{1i} \cdot u_1^{\epsilon_i}, & g^{\rho_i} = U_{2i} \cdot u_2^{\epsilon_i}, & h^{\rho_i} = E_i \cdot (e/g^i)^{\epsilon_i}, \\ (cd^\alpha)^{\rho_i} = V_i \cdot v^{\epsilon_i}, & g_1^{\tau_i} = Z_{1i} \cdot z_{\bar{i}1}^{\epsilon_i}, & g^{\tau_i} = Z_{2i} \cdot (z_{\bar{i}2}/g)^{\epsilon_i}. \end{array}$$

Communication complexity. The receiver message (x_0, x_1, Φ) needs 2+2+4 = 8 group elements. The proof takes 13 elements in $\mathbb{G}$ and 7 elements in $\mathbb{Z}_p$. In particular, the first message has one commitment (i.e., one element in $\mathbb{G}$). The second message has one element[6] in $\mathbb{Z}_p$, and the third messages has 5 elements in $\mathbb{Z}_p$ along with the decommitment (i.e., 12 elements in $\mathbb{G}$ and 1 element in

[6] The second message is in $\{0, 1\}^\lambda$ but we count it as an element of $\mathbb{Z}_p$ for simplicity.

$\mathbb{Z}_p$). The sender message (pk_0, Z_0, pk_1, Z_1) needs $(1, 1, 1, 1) = 4$ group elements in $\mathbb{G}$. Therefore, the total communication complexity amounts to 25 elements in $\mathbb{G}$ and 7 elements in $\mathbb{Z}_p$.

Realizing an adaptively secure channel. Note that the non-committing encryption given in [2] runs in three rounds and needs one public key and one ciphertext of a semantically secure public key encryption scheme. Since the NCE protocol UC-realizes an adaptively secure channel [5, Section 6.3], the NCE protocol messages can be overlapped with the OT protocol messages (aligning the first message of the NCE protocol with the second message of the OT protocol), and thus the final OT protocol runs in four rounds. We can use ElGamal encryption, and the communication overhead amounts to 3 group elements; the public key consists of one element excluding the generator in the CRS, and the ciphertext consists of two elements.

Acknowledgments. We would like to thank the anonymous reviewers for pointing out the need to transmit the sender's messages over an adaptively secure channel, and for additional helpful feedback.

References

1. W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In B. Pfitzmann, editor, *Advances in Cryptology — Eurocrypt 2001*, volume 2045 of *LNCS*, pages 119–135. Springer, 2001.
2. D. Beaver and S. Haber. Cryptographic protocols provably secure against dynamic adversaries. In R. A. Rueppel, editor, *Advances in Cryptology — Eurocrypt '92*, volume 658 of *LNCS*, pages 307–323. Springer, 1992.
3. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology — Crypto 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
4. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *Advances in Cryptology — Crypto 2003*, volume 2729 of *LNCS*, pages 126–144. Springer, 2003.
5. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 136–145. IEEE, 2001.
6. R. Canetti. Obtaining universally compoable security: Towards the bare bones of trust (invited talk). In K. Kurosawa, editor, *Advances in Cryptology — Asiacrypt 2007*, volume 4833 of *LNCS*, pages 88–112. Springer, Dec. 2007.
7. R. Canetti and M. Fischlin. Universally composable commitments. In J. Kilian, editor, *Advances in Cryptology — Crypto 2001*, volume 2139 of *LNCS*, pages 19–40. Springer, 2001.
8. R. Canetti, E. Kushilevitz, and Y. Lindell. On the limitations of universally composable two-party computation without set-up assumptions. *Journal of Cryptology*, 19(2):135–167, Apr. 2006.
9. R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 494–503. ACM Press, May 2002.

10. S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *Advances in Cryptology — Asiacrypt 2009*, volume 5912 of *LNCS*, pages 287–302. Springer, 2009.
11. S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Simple, black-box constructions of adaptively secure protocols. In *6th Theory of Cryptography Conference — TCC 2009*, volume 5444 of *LNCS*, pages 387–402. Springer, 2009.
12. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Advances in Cryptology — Crypto '98*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.
13. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In L. R. Knudsen, editor, *Advances in Cryptology — Eurocrypt 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.
14. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In B. Preneel, editor, *Advances in Cryptology — Eurocrypt 2000*, volume 1807 of *LNCS*, pages 418–430. Springer, 2000.
15. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In J. Kilian, editor, *Advances in Cryptology — Crypto 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, 2001.
16. M. Fischlin, B. Libert, and M. Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In D. H. Lee and X. Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 468–485. Springer, 2011.
17. J. A. Garay, P. MacKenzie, and K. Yang. Efficient and universally composable committed oblivious transfer and applications. In M. Naor, editor, *1st Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 297–316. Springer, Feb. 2004.
18. J. A. Garay, D. Wichs, and H.-S. Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In S. Halevi, editor, *Advances in Cryptology — Crypto 2009*, volume 5677 of *LNCS*, pages 505–523. Springer, 2009.
19. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *Advances in Cryptology — Asiacrypt 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Dec. 2006.
20. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *Advances in Cryptology — Eurocrypt 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
21. S. Halevi and Y. T. Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, 2012.
22. O. Horvitz and J. Katz. Universally-composable two-party computation in two rounds. In A. Menezes, editor, *Advances in Cryptology — Crypto 2007*, volume 4622 of *LNCS*, pages 111–129. Springer, 2007.
23. Y. Ishai, M. Prabhakaran, and A. Sahai. Founding cryptography on oblivious transfer — efficiently. In D. Wagner, editor, *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, 2008.
24. S. Jarecki and V. Shmatikov. Efficient two-party secure computation on committed inputs. In M. Naor, editor, *Advances in Cryptology — Eurocrypt 2007*, volume 4515 of *LNCS*, pages 97–114. Springer, 2007.
25. A. Y. Lindell. Efficient fully-simulatable oblivious transfer. In T. Malkin, editor, *Cryptographers' Track — RSA 2008*, volume 4964 of *LNCS*, pages 52–70. Springer, Apr. 2008.

26. A. Y. Lindell. Adaptively secure two-party computation with erasures. In *Cryptographers' Track — RSA 2009*, LNCS, pages 117–132. Springer, 2009.
27. Y. Lindell. Highly-efficient universally-composable commitments based on the DDH assumption. In *Advances in Cryptology — Eurocrypt 2011*, volume 6632 of *LNCS*, pages 446–466. Springer, 2011.
28. Y. Lindell, E. Oxman, and B. Pinkas. The IPS compiler: Optimizations, variants and concrete efficiency. In *CRYPTO*, pages 259–276, 2011.
29. Y. Lindell and B. Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In *8th Theory of Cryptography Conference — TCC 2011*, volume 6597 of *LNCS*, pages 329–346. Springer, 2011.
30. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457. ACM-SIAM, 2001.
31. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — Eurocrypt '99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.
32. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology — Crypto '91*, volume 576 of *LNCS*, pages 129–140. Springer, 1992.
33. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In D. Wagner, editor, *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.
34. M. Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN. Cryptology ePrint Archive, Report 2002/164, 2002.