On Black-Box Complexity of Universally Composable Security in the CRS model

Carmit Hazay* and Muthuramakrishnan Venkitasubramaniam**

Abstract. In this work, we study the intrinsic complexity of *black-box Universally Composable (UC) secure computation* based on *general assumptions*. We present a thorough study in various corruption modelings while focusing on achieving security in the common reference string (CRS) model. Our results involve the following:

- **Static UC secure computation.** Designing *the first* static UC secure oblivious transfer protocol based on public-key encryption and stand-alone semi-honest oblivious transfer. As a corollary we obtain the first black-box constructions of UC secure computation assuming only two-round semi-honest oblivious transfer.
- **One-sided UC secure computation.** Designing adaptive UC secure two-party computation with single corruptions assuming public-key encryption with oblivious ciphertext generation.
- **Adaptive UC secure computation.** Designing adaptively secure UC commitment scheme assuming only public-key encryption with oblivious ciphertext generation. As a corollary we obtain the first black-box constructions of adaptive UC secure computation assuming only (trapdoor) simulatable public-key encryption (as well as a variety of concrete assumptions).
  We remark that such a result was not known even under non-black-box constructions.

Keywords: UC Secure Computation, Black-Box Constructions, Oblivious Transfer, UC Commitments

1 Introduction

Secure multi-party computation enables a set parties to mutually run a protocol that computes some function f on their private inputs, while preserving a number of security properties. Two of the most important properties are privacy and correctness. The former implies data confidentiality, namely, nothing leaks by the protocol execution but the computed output. The later requirement implies that no corrupted party or parties can cause the output to deviate from the specified function. It is by now well known how to securely compute any efficient functionality [50, 24, 45, 2, 4] in various models and

* Faculty of Engineering, Bar-Ilan University, Israel. Email: carmit.hazay@biu.ac.il. Research partially supported by a grant from the Israel Ministry of Science and Technology (grant No. 3-10883).

** University of Rochester, Rochester, NY 14611, NY. Email: muthuv@cs.rochester.edu. Research supported by Google Faculty Research Grant and NSF Award CNS-1526377

under the stringent simulation-based definitions (following the ideal/real paradigm). Security is typically proven with respect to two adversarial models: the semi-honest model (where the adversary follows the instructions of the protocol but tries to learn more than it should from the protocol transcript), and the malicious model (where the adversary follows an arbitrary polynomial-time strategy), and feasibility results are known in the presence of both types of attacks. The initial model considered for secure computation was of a static adversary where the adversary controls a subset of the parties (who are called corrupted) before the protocol begins, and this subset cannot change. In a stronger corruption model the adversary is allowed to choose which parties to corrupt throughout the protocol execution, and as a function of its view; such an adversary is called adaptive.

These feasibility results rely in most cases on stand-alone security, where a *single* set of parties run a *single* execution of the protocol. Moreover, the security of most cryptographic protocols proven in the stand-alone setting does not remain intact if many instances of the protocol are executed concurrently [40]. The strongest (but also the most realistic) setting for concurrent security is known by *Universally Composable* (UC) security [4]. This setting considers the execution of an unbounded number of concurrent protocols in an arbitrary and adversarially controlled network environment. Unfortunately, stand-alone secure protocols typically fail to remain secure in the UC setting. In fact, without assuming some *trusted help*, UC security is impossible to achieve for most tasks [7, 8, 40]. Consequently, UC secure protocols have been constructed under various *trusted setup* assumptions in a long series of works; see [1, 5, 34, 10, 38, 14] for few examples.

In this work, we are interested in understanding the intrinsic complexity of *UC secure computation*. Identifying the general assumptions required for a particular cryptographic task provides an abstraction of the functionality and the specific hardness that is exploited to obtain a secure realization of the task. The expressive nature of general assumptions allows the use of a large number of concrete assumptions of our choice, even one that may not have been considered at the time of designing the protocols. Constructions that are based on general assumptions are proven in two flavors:

Black-box usage: A construction is black-box if it refers only to the input/output behavior of the underlying primitives.

Non-black-box usage: A construction is non-black box if it uses the code computing the functionality of the underlying primitives.

Typically, non-black-box constructions have been employed to demonstrate feasibility and derive the minimal assumptions required to achieve cryptographic tasks. An important theoretical question is whether or not non-black-box usage of the underlying primitive is necessary in a construction. Besides its theoretical importance, obtaining black-box constructions is related to efficiency as an undesirable effect of non-black-box constructions is that they are typically inefficient and unlikely to be implemented in practice. Fortunately, a recent line of works [32, 26, 47, 25] has narrowed the gap between what is achievable via non-black-box and black-box constructions under minimal assumptions.

More relevant to our context, the work of Ishai, Prabhakaran and Sahai [33] provided the first black-box constructions of UC secure protocols assuming only one-way

functions in a model where all parties have access to an ideal oblivious transfer (OT) functionality. Orthogonally, Choi et al. [12] provided a compiler that transforms any semi-honest OT to a protocol that is secure against malicious static adversaries *in the stand-alone* (i.e. not UC) while assuming that all parties have access to the ideal commitment functionality. In the adaptive setting, the work of Choi et al. provides a transformation from adaptively secure semi-honest oblivious transfer to one that is secure *in the stronger UC setting* against malicious adaptive adversaries while assuming that all parties have access to the ideal commitment functionality. In essence, these works provide black-box constructions, however, they fall short of identifying the necessary minimal general computational assumptions in the UC setting.

Loosely speaking, a UC commitment scheme [7] is a fundamental building block in secure computation which is defined in two phases: in the commit phase a committer commits to a value while keeping it hidden, whereas in the decommit phase the committer reveals the value that it previously committed to. In addition to the standard binding and hiding security properties that any commitment must adhere, commitment schemes that are secure in the UC framework must allow straight-line extraction (where a simulator should be able to extract the content of any valid commitment generated by the adversary) and straight-line equivocation (where a simulator should be able to produce many commitments for which it can later decommit to both 0 and 1). We stress that even security in the static setting requires some notion of equivocation. Due to these rigorous requirements, it has been a real challenge to design black-box constructions of UC secure commitment schemes.

In the context of realizing the UC commitments in the CRS model, Damgård and Nielsen introduced the notion of mixed-commitments in [16]. This construction requires a CRS that is linear in the number of parties and can be instantiated under the N-residuosity and p-subgroup hardness assumptions. In the global CRS model (where a single CRS is introduced for any number of executions), the only known constructions are by Damgård and Groth [15] based on the Strong RSA assumption and Lindell [42] based on the DDH assumption, where the former construction guarantees security in the adaptive setting whereas the later construction provides static security.

Another fundamental building block in secure computation which has been widely studied is oblivious transfer [49, 21]. Semi-honest two-round oblivious transfer can be constructed based enhanced trapdoor permutations [21] and smooth projective hashing [28], and concretely under Discrete Diffie-Hellman (DDH) [46]. Two-round protocols with malicious UC security are presented in the influential paper by Peikert et al. [48] that presents a black-box framework in the common reference string (CRS) model for oblivious transfer, based on dual-mode public-key encryption (PKE) schemes, which can be concretely instantiated under the DDH, quadratic residuosity and Learning with Errors (LWE) hardness assumptions. In a followup work [13], the authors present UC oblivious transfer constructions in the global CRS model assuming DDH, N-residuosity and the Decision Linear Assumption (DLIN). As pointed out in [13], the [48] constructions require a distinct CRS per party. In the context of adaptive UC oblivious transfer protocols, the works of [12] and [22] give constructions in the UC commitment hybrid model where they additionally rely on an assumption that implies adaptive semi-honest oblivious transfer.

It is worth noting that while the works of [48] and [13] provide abstractions of their assumptions, the assumptions themselves are not general enough to help understand the minimal assumptions required to achieve static UC security. In particular, when restricting attention to black-box constructions based on general assumptions, the state-of-the-art literature seems to indicate that achieving UC security in most trusted setup models reduces to constructing two apparently incomparable primitives: *semi-honest oblivious transfer* and *UC commitment schemes*. This leaves the following important question open:

> *What are the minimal (general) assumptions required to construct UC secure protocols, given only* black-box *access to the underlying primitives?*

We note that this question is already well understood in the static setting when relaxing the black-box requirement. Namely, in [18] Damgård, Nielsen and Orlandi showed how to construct UC commitments assuming only semi-honest oblivious transfer in the global CRS model, while additionally assuming a pre-processing phase where the parties participate in a round-robin manner.[1] More recently, Lin, Pass and Venkitasubramaniam [39] improved this result by removing any restricted pre-processing phase. In the same work the authors showed how to achieve UC security in the global CRS model assuming only the existence of semi-honest oblivious transfer. In particular, this construction shows that static UC security can be achieved without assuming UC commitments when relying on non-black-box techniques.

In the stand-alone (i.e. not UC) setting, assuming only the existence of semi-honest oblivious transfer [26, 32, 27] show how to construct secure multiparty computation protocols while relying on the underlying primitives in a black-box manner. More recently, [12] provided black-box constructions that are secure against static adversaries, again, in the stand-alone setting, where all parties have access to an ideal commitment functionality (cf. Proposition 1 in [12]). The latter construction achieves a stronger notion of straight-line simulation, however falls short of achieving static UC security (see more details in Section 3).

In the adaptive setting, the only work that considers a single general assumption that implies adaptive UC security using non-black-box techniques is the result due to Dachman-Soled et al. [14], that shows how to obtain adaptive UC commitments assuming simulatable PKE. Moreover, the best known general assumptions required to achieve black-box UC security are adaptive semi-honest oblivious transfer and UC commitments [17, 12]. Known minimal general assumptions that are required to construct these primitives are (trapdoor) simulatable PKE for adaptive semi-honest oblivious transfer [11] and mixed commitments for UC commitments [17].

1.1 Our Results

In this paper we present a thorough study of black-box UC secure computation in the CRS model; details follow.

[1] In such a pre-processing phase, it is assumed that at most one party is allowed to transmit messages in any round.

Static UC Secure Computation Our first result is given in the static setting, where we demonstrate the feasibility of UC secure computation based on semi-honest oblivious transfer and extractable commitments. More concretely, we prove how to transform any statically semi-honest secure oblivious transfer into one that is secure in the presence of malicious adversaries, giving *only black-box access* to the underlying semi-honest oblivious transfer protocol. Our approach is inspired by the protocols from [27] and [37], where we observe that it is not required to use the full power of static UC commitments. Instead, we employ a weaker primitive that only requires straight-line input extractability. Interestingly, we prove that this weaker notion of security, denoted by extractable commitments [44], can be realized based on any CPA secure PKE. More precisely, we prove the following theorem.

Theorem 11 *(Informally) Assuming the existence of PKE and semi-honest oblivious transfer, then any functionality can be realized in the CRS model with static UC security, where the underlying primitives are accessed in a black-box manner.*

We remark here that this theorem makes a significant progress towards reducing the general assumptions required to construct UC secure protocols. Previously, the only general assumptions based on which we knew how to construct UC secure protocols were mixed-commitments [16] and dual-mode PKE [48] both of which were tailor-made for the particular application. Towards understanding the required minimal assumptions, we recall the work Damgård and Groth in [15] who showed that the existence of UC commitments in the CRS model implies a stand-alone key agreement protocol. Moreover, under black-box constructions, the seminal work of Impagliazzo and Rudich [31] implies that key agreement cannot be based on one-way functions. Thus, there is reasonable evidence to believe that some public-key primitive is required for UC commitments. In that sense, our assumption regarding PKE is close to being optimal. Nevertheless, it is unknown whether the semi-honest oblivious transfer assumption is required.

Our result is shown in two phases. At first we compile the semi-honest oblivious transfer protocol into a new protocol with intermediate security properties in the presence of malicious adversaries. This transformation is an extension of the [27] transformation that is only proven for bit oblivious transfer, whereas our proof works for string oblivious transfer. Next, we use the transformed oblivious transfer protocol in order to construct a maliciously fully secure oblivious transfer. By combining our oblivious transfer with the [33] protocol we obtain a statically generic UC secure computation.

An important corollary is deduced from the work by Gertner et al. [23], who provided a black-box construction of PKE based on any two-round semi-honest oblivious transfer protocol. Specifically, the combination of their result with ours implies the following corollary, which demonstrates that two-round semi-honest oblivious transfer is sufficient in the CRS model to achieve black-box constructions of UC secure protocols.

Corollary 12 *(Informally) Assuming the existence of two-round semi-honest oblivious transfer, then any functionality can be UC realized in the CRS model, where the oblivious transfer is accessed in a black-box manner.*

Implications. In what follows, we make a sequence of interesting observations that are implied by our result in the static UC setting.

- The important result by Canetti, Lindell, Ostrovsky and Sahai [9] presents the first *non-black-box* constructions of static UC secure protocols assuming enhanced trapdoor permutations. In fact, their result can be extended assuming only PKE with oblivious ciphertext generation (which is PKE with the special property that a ciphertext can be obliviously sampled without the knowledge of the plaintext, and can be further realized using enhanced trapdoor permutation). In that sense, our result, assuming PKE with oblivious ciphertext generation, can be viewed as an improvement of [9] when relying on this primitive in a *black-box* manner.
- The pair of works by Damgard, Nielsen and Orlandi [18] and Lin, Pass and Venkitasubramaniam [39] demonstrate that *non-black-box* constructions of UC commitments, and more generally static UC secure computation, can be achieved in the CRS model assuming only semi-honest oblivious transfer. In comparison, our result shows that two-round semi-honest oblivious transfer protocols are sufficient for obtaining *black-box* UC secure computation in the CRS model. Note that most semi-honest oblivious transfer protocols anyway require only two-round of communication, e.g., [21].
- In [38, 39], Lin, Pass and Venkitasubramaniam provided a unified framework for constructing UC secure protocols in any "trusted-setup" model. Their result is achieved by capturing the minimal requirement that implies UC computations in the setup model. More precisely, they introduced the notion of a UC puzzle and showed that any setup model that admits a UC puzzle can be used to securely realize any functionality in the UC setting, while additionally assuming the existence of semi-honest oblivious transfer. Moreover, they showed how to easily construct such puzzles in most models. We remark that our approach can be viewed as providing a framework to construct black-box UC secure protocols in other UC models. More precisely, we show that any setup model that admits the extractable commitment functionality can be used to securely realize any functionality assuming the existence of semi-honest oblivious transfer. In fact, our result easily extends to the chosen key registration authority (KRA) model [1], where it is assumed the existence of a trusted authority that samples public key, secret key pairs for each party, and broadcasts the public key to all parties. We leave it for future work to instantiate our framework in other setup models.
- The fact that our construction only requires PKE and semi-honest oblivious transfer allows an easy translation of static UC security to various efficient implementations under a wide range of concrete assumptions. Specifically, both PKE and (two-round) semi-honest oblivious transfer can be realized under RSA, factoring Blum integers, LWE, DDH, N-residuosity, p-subgroup and coding assumptions. This is compared to prior results that could be based on the later five assumptions [48, 13, 19, 20].
- Recently, Maji, Prabhakaran, and Rosulek [44] initiated the study of the cryptographic complexity of secure computation tasks, while characterizing the relative complexity of a task in the UC setting. Specifically, they established a zero-one law that states that any task is either trivial (i.e., it can be reduced to any other task), or complete (i.e., to which any task can be reduced to), where a functionality $\mathcal{F}$ is said to *reduce* to another functionality $\mathcal{G}$, if there is a UC secure protocol for $\mathcal{F}$ using ideal access to $\mathcal{G}$. More precisely, they showed that assuming the existence of

semi-honest oblivious transfer, every finite two-party functionality is either trivial or complete. While their main theorem relies on the minimal assumption of semi-honest oblivious transfer, their use of the assumption is non-black-box and they leave it as an open problem to achieve the same while relying on oblivious transfer in a black-box manner. Our result makes progress towards establishing this.
In more details, their high-level approach is to identify complete functionalities using four categories, namely, (1) $\mathcal{F}_{\mathrm{XOR}}$ that abstracts a XOR-type functionality, (2) $\mathcal{F}_{\mathrm{CC}}$ that abstracts a simple cut-and-choose functionality, (3) $\mathcal{F}_{\mathrm{OT}}$ the oblivious transfer functionality, and (4) $\mathcal{F}_{\mathrm{COM}}$ the commitment functionality. They then show that each category can be used to securely realize any computational task.[2] Among these reductions, functionalities $\mathcal{F}_{\mathrm{XOR}}$ and $\mathcal{F}_{\mathrm{CC}}$ rely on oblivious transfer in a non-black-box way. In this work we improve the reduction of functionality $\mathcal{F}_{\mathrm{CC}}$. That is, we obtain this improvement by showing that the extractable commitment functionality $\mathcal{F}_{\mathrm{EXTCOM}}$ and semi-honest oblivious transfer can be used in a black-box way to realize functionality $\mathcal{F}_{\mathrm{OT}}$, and combine this with a reduction presented in [44] that reduces $\mathcal{F}_{\mathrm{CC}}$ to the $\mathcal{F}_{\mathrm{EXTCOM}}$ functionality in a black-box way.

One-Sided UC Secure Computation In this stronger two-party setting, where at most one of the parties is adaptively corrupted [35, 29], we prove that one-sided adaptive UC security is implied by PKE with oblivious ciphertext generation. Here we combine two observations, one where our malicious static oblivious transfer from the previous result requires using the parties' inputs in only one phase, together with the fact that one-sided non-committing encryption (NCE) can be designed based on PKE with oblivious ciphertext generation [6, 16]. In particular, NCE allow secure communication in the presence of adaptive attacks, which implies that the communication can be equivocated once the real message is handed to the simulator. Then, by encrypting part of our statically secure protocol using NCE, we obtain a generic protocol for any two-party functionality under the assumption specified above.[3] Namely,

Theorem 13 *(Informally) Assuming the existence of PKE with oblivious ciphertext generation, then any two-party functionality can be realized in the CRS model with one-sided adaptive UC security and black-box access to the PKE.*

Adaptive UC Secure Computation Our last result is in the strongest corruption setting, where any number of parties can be adaptively corrupted. Here we design a new adaptively secure UC commitment scheme under the assumption of PKE with oblivious ciphertext generation, which is the first construction that achieves the stronger notion of adaptive security based on this hardness assumption. Our construction makes a novel usage of such a PKE together with Reed-Solomon codes, where the polynomial shares are encrypted using the PKE with oblivious ciphertext generation. Plugging-in our UC commitment protocol into the transformation of [12] that generates adaptive malicious

[2] Where it suffices to realize the $\mathcal{F}_{\mathrm{OT}}$ functionality as it is known to be complete [36].

[3] We note that while in the plain model any statically secure protocol can be compiled into one-sided secure protocol by encrypting its entire communication using one-sided NCE, it is not the case in the UC setting due to the additional setup.

oblivious transfer given adaptive semi-honest oblivious transfer and UC commitments, implies an adaptively UC secure oblivious transfer protocol with malicious security based on semi-honest adaptive oblivious transfer and PKE with oblivious ciphertext generation, using only black-box access to the semi-honest oblivious transfer and the PKE. That is,

Theorem 14 *(Informally) Assuming the existence of PKE with oblivious ciphertext generation and adaptive semi-honest oblivious transfer, then any functionality can be realized in the CRS model with adaptive UC security, where the underlying primitives are accessed in a black-box manner.*

We further recall the work of Choi et al. [11] that shows that the weakest general known assumption that is required to construct adaptively secure semi-honest oblivious transfer is trapdoor simulatable PKE. Now, since such an encryption scheme admits PKE with oblivious ciphertext generation, we obtain the following corollary that unifies the two assumptions required to achieve adaptive UC security.

Corollary 15 *Assuming the existence of (trapdoor) simulatable PKE, then any functionality can be realized in the CRS model with adaptive UC security and black-box access to the PKE.*

An additional interesting observation that is implied by our work is that our UC commitment scheme implies a construction that is secure in the adaptive setting when erasures are allowed, and under the weaker assumption of PKE. Specifically, instead of obliviously sampling ciphertexts in the commitment phase, the committer encrypts arbitrary plaintexts and then erases the plaintexts and randomness used for these computations. Our proof follows easily for this case as well. Combining our UC commitment scheme together with the semi-honest with erasures OT from [41] and the transformation of [12], we obtain the following result

Theorem 16 *(Informally) Assuming the existence of PKE and semi-honest oblivious transfer secure against an adaptive adversary assuming erasures, then any functionality can be realized in the CRS model with adaptive UC security assuming erasures, where the underlying primitives are accessed in a black-box manner.*

Noting that OT secure against adaptive adversaries assuming erasures can be realized under assumptions sufficient for achieving the same with respect to the weaker static adversaries, this theorem shows that achieving UC security against adaptive adversaries in the presence of erasures does not require any additional assumption beyond what is required to secure against static adversaries.

Implications. Next, we specify a sequence of interesting observations that are implied by our result in the adaptive UC setting.

- Previously, Dachman-Soled et al. [14], showed that adaptive UC secure protocols can be constructed in the CRS model assuming the existence of simulatable PKE. Our result improves this result in terms of complexity assumptions by showing that trapdoor simulatable PKE is sufficient, and provides new constructions based on concrete assumptions that were not known before. Nevertheless, we should point

out that while the work of Dachman-Soled et al. is constructed in the global CRS model using a non-black-box construction, our result provides a black-box construction in a CRS model where the length of the reference string is linear in the number of parties.
- Analogous to our result on static UC security, it is possible to extend this result to the chosen key-registration authority (KRA) model, where we assume the existence of a trusted-party that samples public keys and secret keys for each party, and broadcasts the public key to all parties.
- Importantly, this result provides the first evidence that adaptively secure UC commitment is theoretically easier to construct than stand-alone adaptively secure semi-honest oblivious transfer. This is due to a separation from [43] (regarding static vs. adaptive oblivious transfer), that proves that adaptive oblivious transfer requires a stronger hardness assumption than enhanced trapdoor permutation.
- Regarding concrete assumptions, previously, adaptive UC commitments without erasures were constructed based on N-residuosity and p-subgroup hardness assumptions [17] and Strong RSA [15]. On the other hand, our result demonstrates the feasibility of this primitive under DDH, LWE, factoring Blum integers and RSA assumptions. When considering adaptive corruption with erasures, the work of Blazy, et al. [3], extending the work of Lindell [42], shows how to construct highly efficient UC commitments based on the DDH assumption. On the other hand, assuming erasures, we are able to construct an adaptive UC commitment scheme based on any CPA-secure PKE.

2 Preliminaries

We denote the security parameter by n. We use the abbreviation PPT to denote probabilistic polynomial-time. We further denote by $a \leftarrow A$ the random sampling of a from a distribution A, and by $[n]$ the set of elements $\{1, \ldots, n\}$.

Definition 21 (PKE with oblivious ciphertext generation [16]) *A PKE Π with oblivious sampling generation is defined by the tuple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Enc}}^{-1})$ *and has the following additional property,*

- **Indistinguishability of oblivious and real ciphertexts.** *For any message m in the appropriate domain, consider the experiment* $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^n)$, $c_1 \leftarrow \widetilde{\mathsf{Enc}}_{\mathrm{PK}}(r_1)$, $c_2 \leftarrow \mathsf{Enc}_{\mathrm{PK}}(m; r_2)$, $r_1' \leftarrow \widetilde{\mathsf{Enc}}^{-1}_{\mathrm{PK}}(c_2)$.
  Then, $(\mathrm{PK}, r_1', c_1, m) \overset{\mathrm{c}}{\approx} (\mathrm{PK}, r_2, c_2, m)$.

To this end, we only employ PKE with perfect decryption. This merely simplifies the analysis and can be relaxed by using PKE with a negligible decryption error instead.

2.1 Oblivious Transfer

1-out-of-2 oblivious transfer (OT) is an important functionality in the context of secure computation that is engaged between a sender Sen and a receiver Rec; see Figure 1

Functionality $\mathcal{F}_{\mathrm{OT}}$

Functionality $\mathcal{F}_{\mathrm{OT}}$ communicates with with sender Sen and receiver Rec, and adversary $\mathcal{S}$.

1. Upon receiving input $(\mathsf{sender}, sid, v_0, v_1)$ from Sen where $v_0, v_1 \in \{0,1\}^t$, record (sid, v_0, v_1).
2. Upon receiving $(\mathsf{receiver}, sid, u)$ from Rec, where a tuple (sid, v_0, v_1) is recorded and $u \in \{0,1\}$, send (sid, v_u) to Rec and sid to $\mathcal{S}$. Otherwise, abort.

Fig. 1. The oblivious transfer functionality.

for the description of functionality $\mathcal{F}_{\mathrm{OT}}$. In this paper we are interested in reducing the hardness assumptions for general UC secure computation when using only black-box access to the underlying cryptographic primitives, such as the semi-honest OT. We use semi-honest OT as a building block for designing UC secure protocols in both static and adaptive settings. In the static setting, we refer to the two-round protocol of [21] that is based on PKE with oblivious ciphertext generation (or enhanced trapdoor permutation). In the adaptive setting, we refer to the two-round protocol of [9] that is based on augmented non-committing encryption scheme.

We next recall that any two-round semi-honest OT implies PKE. We demonstrate that in two phases, starting with the claim that semi-honest OT implies a key agreement (KA) protocol, where two parties agree on a secret key over a public channel. This statement has already been proven in [23] in the static setting, and holds for any number of rounds. The idea is simple, the parties execute an OT protocol where the party that plays the sender picks two random inputs s_0, s_1, whereas the party that plays the receiver enters 0. Finally, the parties output s_0 and security follows from the correctness and privacy of the OT. A simple observation shows that this reduction also holds in the adaptive setting. Namely, starting with an adaptive semi-honest OT, the same reduction implies an adaptively secure KA (where the protocol communication must be consistent with respect to any key). Note that this reduction preserves the number of rounds, thus if the starting point is a two-round OT then the reduction implies a two-round KA. Next, a well established fact shows that in the static setting a two-round key agreement implies PKE (in fact, these primitives are equivalent). Formally,

Theorem 22 *Assume the existence of two-round key agreement protocol with static security, then there exists IND-CPA PKE.*

Sender Private Oblivious Transfer Sender privacy is a weaker notion than malicious security and only requires that the receiver's input be hidden even against a malicious sender. It is weaker than malicious security in that it does not require a simulation of the malicious sender that extracts the sender's inputs. In particular, we will only require that a malicious sender cannot distinguish the cases where the receiver's input is 0 or 1. Formally stated,

Definition 23 (Sender private OT) *Let π be a two-party protocol that is engaged between a sender* Sen *and a receiver* Rec*. We say that π is a* sender private *oblivious transfer protocol, if for every* PPT *adversary $\mathcal{A}$ that corrupts* Sen*, the following ensembles are computationally indistinguishable:*

- $\{\mathbf{View}_{\mathcal{A},\pi}[\mathcal{A}(1^n), \mathrm{Rec}(1^n, 0)]\}_{n\in\mathbb{N}}$
- $\{\mathbf{View}_{\mathcal{A},\pi}[\mathcal{A}(1^n), \mathrm{Rec}(1^n, 1)]\}_{n\in\mathbb{N}}$

where $\mathbf{View}_{\mathcal{A},\pi}[\mathcal{A}(1^n), \mathrm{Rec}(1^n, b)]$ *denotes $\mathcal{A}$'s view within π whenever the receiver* Rec *inputs the bit b.*

We point out that sender privacy protects the receiver against a malicious sender and should be read as privacy against a malicious sender.

Defensibly Private Oblivious Transfer The notion of *defensible privacy* was introduced by Haitner in [26, 27]. A defense in a two-party protocol $\pi = (P_1, P_2)$ execution is an input and random tape provided by the adversary after the execution concludes. A defense for a party controlled by the adversary is said to be *good*, if this party participated honestly in the protocol using this very input and random tape, then it would have resulted in the exact same messages that were sent by the adversary. In essence, this defense serves as a *proof* of honest behavior. It could very well be the case that an adversary deviates from the protocol in the execution but later provides a good defense. The notion of defensible privacy says that a protocol is private in the presence of defensible adversaries if the adversary learns nothing more than its prescribed output when it provides a good defense.

We informally describe the notion of *good defense* for a protocol π; we refer to [27] for the formal definition. Let $\mathsf{trans} = (q_1, a_1, \ldots, q_\ell, a_\ell)$ be the transcript of an execution of a protocol π that is engaged between P_1 and P_2 and let $\mathcal{A}$ denote an adversary that controls P_1, where q_i is the ith message from P_1 and a_i is the ith message from P_2 (that is, a_i is the response for q_i). Then we say that (x, r) constitutes a *good defense* of $\mathcal{A}$ relative to trans if the transcript generated by running the honest algorithm for P_1 with input x and random tape r against P_2's messages $a_1, \ldots, a_\ell$ results trans.

The notion of defensible privacy can be defined for any secure computation protocol. Nevertheless, since we are only interested in oblivious transfer protocols, we present a definition below that is restricted to oblivious transfer protocols. The more general definition can be found in [27]. At a high-level, an OT protocol is defensibly private with respect to a corrupted sender if no adversary interacting with an honest receiver with input b should be able to learn b, if at the end of the execution the adversary produces any good defense. Similarly, an OT protocol that is defensibly private with respect to malicious receivers requires that any adversary interacting with an honest sender with input (s_0, s_1) should not be able to learn s_{1-b}, if at the end of the execution the adversary produces a good defense with input b. Below we present a variant of the definition presented in [27]. We stress that while the [27] definition only considers bit OT (i.e. sender's inputs are bits) we consider *string OT*.

Definition 24 (Defensible-private string OT) *Let π be a two-party protocol that is engaged between a sender* Sen *and a receiver* Rec*. We say that π is a* defensibly-private *string oblivious transfer protocol, if for every* PPT *adversary $\mathcal{A}$ the following holds,*

1. $\{\Gamma(\mathbf{View}_{\mathcal{A}}[\mathcal{A}(1^n), \mathrm{Rec}(1^n, U)], U)\} \overset{c}{\approx} \{\Gamma(\mathbf{View}_{\mathcal{A}}[\mathcal{A}(1^n), \mathrm{Rec}(1^n, U)], U')\}$, *where* $\Gamma(v, *)$ *is set to* $(v, *)$ *if following the execution* $\mathcal{A}$ *outputs a good defense for* π*, and* $\perp$ *otherwise, and* U *and* U' *are independent random variables uniformly distributed over* $\{0,1\}$*. This property is referred to as defensibly private with respect to a corrupted sender.*
2. $\{\Gamma(\mathbf{View}_{\mathcal{A}}[\mathrm{Sen}(1^n, (U_0^n, U_1^n)), \mathcal{A}(1^n)], U_{1-b}^n)\} \overset{c}{\approx} \{\Gamma(\mathbf{View}_{\mathcal{A}}[\mathrm{Sen}(1^n, (U_0^n, U_1^n)), \mathcal{A}(1^n)], \bar{U}^n)\}$ *where* $\Gamma(v, *)$ *is set to* $(v, *)$ *if following the execution* $\mathcal{A}$ *outputs a good defense for* π*, and* $\perp$ *otherwise,* b *is the* Rec*'s input in this defense and* $U_0^n, U_1^n, \bar{U}^n$ *are independent random variables uniformly distributed over* $\{0,1\}^n$*. This property is referred to as defensibly private with respect to a corrupted receiver.*

In our construction from Section 3, we will rely on an OT protocol that is sender private and defensibly private with respect to a corrupted receiver. In [27], Haitner et al. showed how to transform any semi-honest bit-OT to one that is defensibly private with respect to a corrupted receiver and malicious secure with respect to a corrupted sender. More formally, the following Lemma is implicit in the work of [27].

Lemma 21 (Implicit in Theorem 4.1 and Corollary 5.3 [27]) *Assume the existence of a semi-honest oblivious transfer protocol* π*. Then there exists an oblivious transfer protocol* $\hat{\pi}$ *that is defensible-private with respect to the receiver and sender private that relies on the underlying primitive in a black-box manner.*

Now, since sender privacy is implied by malicious security with respect to a corrupted sender, this transformation yields a bit OT protocol with the required security guarantees. Nevertheless, our protocol crucially relies on the fact that the underlying OT is a string OT protocol. We therefore show in the full version [30] how to transform any bit OT to a string OT protocol while preserving both defensible private with respect to a maliciously corrupted receiver and sender privacy.

At a high-level, in order to convert any protocol from semi-honest security to defensible privacy, Haitner et al. include a coin-tossing stage at the beginning of the protocol that determines the parties' random tapes. In fact, they let the coin-tossing also determine the parties inputs as they only require OT secure with respect to random inputs for both the sender and receiver. Now, if the receiver has to provide a good defense, then it must reveal the input and randomness used for the semi-honest OT protocol and prove consistency relative to the values generated in the coin-tossing stage. Due to the fact that the commitment schemes that are used in the coin-tossing stage are statistically-binding, the probability that a malicious receiver can deviate from the protocol and provide a good defense is negligible. Using this fact, Haitner et al. argued that the probability that a malicious receiver outputs a good defense and guesses the other sender's input is negligible. Next, to obtain sender private oblivious transfer they first transformed an OT protocol that is defensible-private against malicious receivers to one that is maliciously secure, and then exploited the symmetry of OT in order to obtain a protocol that is sender-private. The first transformation relies on the cut-and-choose approach to ensure that the receiver provides a valid defense, and then using the fact that defensible privacy hides the sender's other input they argued that it is receiver-private.

Functionality $\mathcal{F}_{\mathrm{COM}}$

Functionality $\mathcal{F}_{\mathrm{COM}}$ communicates with with sender Sen and receiver Rec, and adversary $\mathcal{S}$.

1. Upon receiving input $(\mathsf{commit}, sid, m)$ from Sen where $m \in \{0,1\}^t$, internally record (sid, m) and send message $(sid, \mathrm{Sen}, \mathrm{Rec})$ to the adversary. Upon receiving $\mathsf{approve}$ from the adversary send sid, to Rec. Ignore subsequent $(\mathsf{commit}, ., ., .)$ messages.
2. Upon receiving (reveal, sid) from Sen, where a tuple (sid, m) is recorded, send message m to adversary $\mathcal{S}$ and Rec. Otherwise, ignore.

Fig. 2. The string commitment functionality.

2.2 UC Commitment Schemes

The notion of UC commitments was introduced by Canetti and Fischlin in [7]. The formal description of functionality $\mathcal{F}_{\mathrm{COM}}$ is depicted in Figure 2.

2.3 Extractable Commitments

Our result in the static setting requires the notion of (static) extractable UC commitments, which is a weaker security property than UC commitments in the sense that it does not require equivocality. In what follows, we introduce the definition for the ideal functionality $\mathcal{F}_{\mathrm{EXTCOM}}$ from [44]. Towards introducing this definition, Maji et al. introduced some notions first. More concretely,

Definition 25 *A protocol is a* syntactic commitment protocol *if:*

- *It is a two phase protocol between a sender and a receiver (using only plain communication channels).*
- *At the end of the first phase (commitment phase), the sender and the receiver output a transcript* trans*. Furthermore, the sender receives an output (which will be used for opening the commitment).*
- *In the decommitment phase the sender sends a message γ to the receiver, who extracts an output value* $\mathsf{opening}(\mathsf{trans}, \gamma) \in \{0,1\}^n \cup \{\perp\}$.

Definition 26 *Two syntactic commitment protocols (ω_L, ω_R) form a pair of complementary statistically binding commitment protocols if the following hold:*

- *ω_R is a statistically binding commitment scheme (with stand-alone security).*
- *In ω_L, at the end of the commitment phase the receiver outputs a string $z \in \{0,1\}^n$. If the receiver is honest, it is only with negligible probability that there exists γ such that* $\mathsf{opening}(\mathsf{trans}, \gamma) \neq \perp$ *and* $\mathsf{opening}(\mathsf{trans}, \gamma) \neq z$.

As noted in [44], ω_L by itself is not an interesting cryptographic goal, as the sender can simply send the committed string in the clear during the commitment phase. Nevertheless, in defining $\mathcal{F}_{\text{EXTCOM}}$ below, there exists a single protocol that satisfies both the security guarantees. We are now ready to introduce the notion of extractable commitments in Figure 3 that is parameterized by (ω_L, ω_R). We also include a function pp that will be used as an initialization phase to set up the public-parameters for ω_L and ω_R.

Functionality $\mathcal{F}_{\text{EXTCOM}}$ parameterized by $(\mathsf{pp}, \omega_L, \omega_R)$

$\mathcal{F}_{\text{EXTCOM}}$ is running with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$: Upon receiving a message $(\mathsf{init-commit}, sid, ssid, P_i, P_j)$ from P_i, it first checks if there is a tuple $(\mathsf{public-params}, sid, P_i, (pp, sp))$. If yes, it sends $(\mathsf{init-commit}, sid, ssid, P_i, P_j)$ to P_j. If not, it runs $(pp, sp) \leftarrow \mathsf{pp}(1^n)$ and sends $(\mathsf{init-commit}, sid, P_i, pp)$ to P_i, P_j and $\mathcal{S}$. It stores $(\mathsf{public-params}, sid, P_i, (pp, sp))$. We denote P_i by the sender and P_j by the receiver in this interaction. Next, the functionality behaves as follows, depending on which party is corrupted.

- P_i IS HONEST AND P_j IS HONEST.
  - **Commit Phase:** Upon receiving $(\mathsf{commit}, sid, ssid, P_i, P_j, m)$ from P_i, it internally simulates a session of ω_R (simulating both the sender and receiver in ω_R), with the sender's input fixed to m. It gives $(\mathsf{transcript}, sid, ssid, \mathsf{trans}, \gamma)$ to P_i and $(\mathsf{receipt}, sid, ssid, P_i, P_j, \mathsf{trans})$ to P_j and $\mathcal{S}$.
  - **Reveal Phase:** Upon receiving $(\mathsf{decommit}, sid, ssid, \cdot)$ from the sender, it sends $(\mathsf{decommit}, sid, ssid, P_i, P_j, z)$ to P_j and $\mathcal{S}$.
- P_i IS CORRUPTED AND P_j IS HONEST.
  - **Commit Phase:** It runs the commitment ω_L with the sender, playing the part of the receiver in ω_L, to obtain $(sid, ssid, \mathsf{trans}, z)$. It sends $(\mathsf{receipt}, sid, ssid, P_i, P_j, \mathsf{trans})$ to P_j and $\mathcal{S}$.
  - **Reveal Phase:** Upon receiving $(\mathsf{decommit}, sid, ssid, \gamma)$ from the sender, if $\mathsf{opening}(\mathsf{trans}, \gamma) = z$, it sends $(\mathsf{decommit}, sid, ssid, P_i, P_j, z)$ to P_j and $\mathcal{S}$. Otherwise ignore.
- P_i IS HONEST AND P_j IS CORRUPT.
  - **Commit Phase:** Upon receiving $(\mathsf{commit}, sid, ssid, P_i, P_j, m)$ from P_i, it runs the commitment phase of ω_R with P_j, playing the sender's role in ω_R with m as input. It obtains the output (trans, γ) at the end of this phase, and sends $(\mathsf{transcript}, sid, ssid, \mathsf{trans}, \gamma)$ to P_i.
  - **Reveal Phase:** Upon receiving $(\mathsf{decommit}, sid, ssid)$ from the sender it sends $(\mathsf{decommit}, sid, ssid, P_i, P_j, (\gamma, z))$ to P_j and $\mathcal{S}$.

The functionality does not do anything when both the sender and the receiver are corrupted.

Fig. 3. Extractable commitment functionality.

Implementing $\mathcal{F}_{\text{EXTCOM}}$ in the CRS model. We briefly sketch how to implement the extractable commitment functionality in the $\mathcal{F}_{\text{CRS}}$-hybrid based on the CPA-security of any PKE. Namely, the CRS will be set to a public-key generated using the key-

generation function of the PKE scheme. To commit, a sender simply encrypts the message using the public-key in the CRS and sends the ciphertext to the receiver. We can achieve extraction by setting the CRS to a public-key for which the secret-key is available to the extractor (in this case, the extractor is the $\mathcal{F}_{\text{EXTCOM}}$ functionality). Hiding follows from the CPA-security of the encryption scheme. A formal description and proof of this construction can be found in the full version of this paper [30].

3 Static UC Secure Computation

In this section we prove the feasibility of UC secure computation based on semi-honest OT and extractable commitments, where the latter can be constructed based on two-round semi-honest OT (see Sections 2.1 and 2.3 for more details). More concretely, we prove how to transform any statically semi-honest secure OT into one that is secure in the presence of malicious adversaries, giving *only black-box access* to the underlying semi-honest OT protocol. Our protocol is a variant of the protocol by Lin and Pass from [37] (which in turn is a variant of the protocol of [27]). In particular, in [37], the authors rely on a strong variant of a commitment scheme known as a CCA-secure commitment in order to achieve extraction. We observe that it is not required to use the full power of such commitments, or for that matter UC commitments. Specifically, using a weaker primitive that only implies straight-line input extractability enables to solely rely on semi-honest OT. An important weakening in our commitment scheme compared to CCA-secure commitments from [37] is that we allow invalid commitments to be made by the adversary. We remark here that the work of [37] rely on string OT that are secure against malicious senders and state that the work of [26] provides a black-box construction of such a protocol starting from a semi-honest bit OT. However, the work of [26] only shows how to construct a bit OT secure against malicious senders where the proof crucially relies on the sender's input being only bits. We provide a transformation and complete analysis from bit OT to a string OT for the weaker notion of defensible privacy as this is sufficient for our work. Finally, combining our UC OT protocol with the [33] protocol, we obtain a statically UC secure protocol for any well-formed functionality (see definition in [9]). Namely,

Theorem 31 *Assume the existence of static semi-honest oblivious transfer. Then for any multi-party well-formed functionality $\mathcal{F}$, there exists a protocol that UC realizes $\mathcal{F}$ in the presence of static, malicious adversaries in the $\mathcal{F}_{\text{EXTCOM}}$-hybrid model using black-box access to the oblivious transfer protocol.*

We remark here that the work of [12] shows how starting from a semi-honest oblivious transfer it is possible to obtain a black-box construction of an OT protocol that is secure against stand-alone static adversaries in the $\mathcal{F}_{\text{COM}}$-hybrid model. It is noted in [12] that the (high-level) analysis provided in the work might be extendable to the UC-setting (cf. Footnote 10 in [12]). Furthermore, in the static setting, it is conceivable that $\mathcal{F}_{\text{COM}}$ can be directly realized in the $\mathcal{F}_{\text{EXTCOM}}$-hybrid using the notion of extractable trapdoor commitments [47]. We do not pursue this approach and instead directly realize OT in the $\mathcal{F}_{\text{EXTCOM}}$-hybrid. While the previous works of [12] and [27] require a three step transformation, our transformation is one shot and therefore more direct.

It seems possible to generalize our theorem to multi-session functionalities. Analogous to [7], this will allows us to extend our corollaries to the Global CRS model by additionally assuming CCA encryption scheme and leave it as future work.

3.1 Static UC Oblivious Transfer

In the following, we discuss a secure implementation of the oblivious transfer functionality (see Figure 1) with static, malicious security in the $\mathcal{F}_{\text{EXTCOM}}$-hybrid model (where $\mathcal{F}_{\text{EXTCOM}}$ is stated formally in Figure 3). Our goal in this section is to show that the security of malicious UC OT can be based on UC semi-honest OT, denoted by $\pi_{\text{OT}}^{\text{SH}}$, and extractable commitments. Our result is shown in two phases. At first we compile the semi-honest OT protocol $\pi_{\text{OT}}^{\text{SH}}$ into a new protocol with the security properties that are specified in Section 2.1, extending the [27] transformation into string OT; denote the compiled OT protocol by $\widehat{\pi}_{\text{OT}}$. Next, we use $\widehat{\pi}_{\text{OT}}$ in order to construct a new protocol $\pi_{\text{OT}}^{\text{ML}}$ that is secure in the presence of malicious adversaries. Details follow,

Protocol 1 (Protocol $\pi_{\text{OT}}^{\text{ML}}$ with static security)
Input: *The sender* Sen *has input* (v_0, v_1) *where* $v_0, v_1 \in \{0,1\}^n$ *and the receiver* Rec *has input* $u \in \{0,1\}$.
The protocol:

1. **Coin tossing:**
   - Receiver's random tape generation: *The parties use a coin tossing protocol in order to generate the inputs and random tapes for the receiver.*
     - *The receiver commits to* $20n$ *strings of appropriate length, denoted by* $a_{\text{Rec}}^1, \ldots, a_{\text{Rec}}^{20n}$, *by sending* $\mathcal{F}_{\text{EXTCOM}}$ *the message* $(\mathsf{commit}, sid, \widetilde{ssid}_i, a_{\text{Rec}}^i)$ *for all* $i \in [n]$.
     - *The sender responds with* $20n$ *random strings of appropriate length* $b_{\text{Rec}}^1, \ldots, b_{\text{Rec}}^{20n}$.
     - *The receiver computes* $r_{\text{Rec}}^i = a_{\text{Rec}}^i \oplus b_{\text{Rec}}^i$ *and then interprets* $r_{\text{Rec}}^i = c_i || \tau_{\text{Rec}}^i$ *where* c_i *determines the receiver's input for the* i^{th} *OT protocol, whereas* τ_{Rec}^i *determines the receiver's random tape used for this execution.*
   - Sender's random tape generation: *The parties use a coin tossing protocol in order to generate the inputs and random tapes for the sender.*
     - *The sender commits to* $20n$ *strings of appropriate length, denoted by* $a_{\text{Sen}}^1, \ldots, a_{\text{Sen}}^{20n}$, *by sending* $\mathcal{F}_{\text{EXTCOM}}$ *the message* $(\mathsf{commit}, sid, \widetilde{ssid}'_i, a_{\text{Sen}}^i)$ *for all* $i \in [n]$.
     - *The receiver responds with* $20n$ *random strings of appropriate length* $b_{\text{Sen}}^1, \ldots, b_{\text{Sen}}^{20n}$.
     - *The sender computes* $r_{\text{Sen}}^i = a_{\text{Sen}}^i \oplus b_{\text{Sen}}^i$ *and then interprets* $r_{\text{Sen}}^i = s_i^0 || s_i^1 || \tau_{\text{Sen}}^i$ *where* (s_i^0, s_i^1) *determine the sender's input for the* i^{th} *OT protocol, whereas* τ_{Sen}^i *determines the sender's random tape used for this execution.*
2. **Oblivious transfer:**
   - *The parties participate in* $20n$ *executions of the OT protocol* $\widehat{\pi}_{\text{OT}}$ *with the corresponding inputs and random tapes obtained from Stage 2. Let the output of the receiver in the* i^{th} *execution be* $\tilde{s}_i$.
3. **Cut-and-choose:**
   - Sen *chooses a random subset* $q_{\text{Sen}} = (q_{\text{Sen}}^1, \ldots, q_{\text{Sen}}^n) \in \{1, \ldots, 20\}^n$ *and sends it to* Rec. *The string* q_{Sen} *is used to define a set of indices* $\Gamma_{\text{Sen}} \subset \{1, \ldots, 20n\}$ *of size* n *in the following way:* $\Gamma_{\text{Sen}} = \{20i - q_{\text{Sen}}^i\}_{i \in [n]}$. *The receiver then opens the commitments from Stage 1 that correspond to the indices within* Γ_{Sen}, *namely, the receiver decommits* a_{Rec}^i *for all* $i \in \Gamma_{\text{Sen}}$. Sen *checks that the decommitted values are consistent with the inputs and randomness used for the OTs in Stage 2 by the receiver, and aborts in case of a mismatch.*

- Rec *chooses a random subset* $q_{\mathrm{Rec}} = (q^1_{\mathrm{Rec}}, \ldots, q^n_{\mathrm{Rec}}) \in \{1, \ldots, 20\}^n$ *and sends it to* Sen. *The string* q_{Rec} *is used to define a set of indices* $\Gamma_{\mathrm{Rec}} \subset \{1, \ldots, 20n\}$ *of size* n *in the following way:* $\Gamma_{\mathrm{Rec}} = \{20i - q^i_{\mathrm{Rec}}\}_{i \in [n]}$. *The sender then opens the commitments from Stage 1 that correspond to the indices within* Γ_{Rec}*, namely, the sender decommits* a^i_{Sen} *for all* $i \in \Gamma_{\mathrm{Rec}}$. Rec *checks that the decommitted values are consistent with the inputs and randomness used for the OTs in Stage 2 by the sender, and aborts in case of a mismatch.*
- Rec *commits to another subset* $\Gamma \subset [20n]$ *denoted by* $(\Gamma^1, \ldots, \Gamma^n)$*, by sending* $\mathcal{F}_{\mathrm{EXTCOM}}$ *the message* $(\mathsf{commit}, sid, ssid'_i, \Gamma^i)$ *for all* $i \in [n]$. *(The sender will reveal its inputs and randomness that are used in Stage 2 that correspond to the indices in* Γ *later in Stage 5.)*

4. **Combiner:**
   - *Let* $\Delta = [20n] - \Gamma_{\mathrm{Rec}} - \Gamma_{\mathrm{Sen}}$. *Then for every* $i \in \Delta$*, the receiver computes* $\alpha_i = u \oplus c_i$ *and sends it to the sender.*
   - *The sender computes a* $10n$*-out-of-*$18n$ *secret sharing of* v_0*, denote the shares by* $\{\rho^0_i\}_{i \in \Delta}$. *Analogously, it computes a* $10n$*-out-of-*$18n$ *secret sharing of* v_1*, denote the shares by* $\{\rho^1_i\}_{i \in \Delta}$. *The sender computes* $\beta^b_i = \rho^b_i \oplus s_i^{b \oplus \alpha_i}$ *for all* $b \in \{0, 1\}$ *and* $i \in \Delta$*, and sends the outcome to the receiver.*
   - *The receiver computes* $\tilde{\rho}_i = \beta^u_i \oplus \tilde{s}_i$ *for all* $i \in \Delta$. *Denote by* ρ *these concatenated bits.*
5. **Final cut-and-choose:**
   - *The receiver decommits* Γ *and the sender sends the inputs and randomness it used in Stage 2 for the coordinates that correspond to* $\Delta \cap \Gamma$. *(Note that the sender need only reveal the indices that were not decommitted in Stage 3).* Rec *checks that the sender's values are consistent with the inputs and randomness used for the OTs in Stage 2 by the sender, and aborts in case of a mismatch.*
   - *The receiver checks whether* $(\tilde{\rho}_i)_{i \in \Delta}$ *agrees with some codeword* $w \in \mathcal{W}_{18n,10n}$ *on* $17n$ *locations (where the code* $\mathcal{W}_{18n,10n}$ *is induced by the secret sharing construction that we use in Stage 4). Recall that the minimum distance of the code* $\mathcal{W}_{18n,10n}$ *is at least* $18n - 10n > 8n$*, which implies that there will be at most one such codeword* w*. Furthermore, since we can correct up to* $\frac{18n-10n}{2} = 4n$ *errors, any code that is* $17n$ *close to a codeword can be efficiently recovered using the Berlekamp-Welch algorithm. The receiver outputs that* w *as its output in the OT protocol. If no such* w *exists, the receiver returns a default value.*

Theorem 32 *Assume that* $\pi^{\mathrm{SH}}_{\mathrm{OT}}$ *is static semi-honest secure and that the compiled* $\widehat{\pi}_{\mathrm{OT}}$ *is secure according to Lemma 21. Then Protocol 1 UC realizes* $\mathcal{F}_{\mathrm{OT}}$ *in the presence of static malicious adversaries in the* $\mathcal{F}_{\mathrm{EXTCOM}}$*-hybrid model using black-box access to the oblivious transfer protocol.*

Recalling that our protocol relies on the existence of semi-honest OT and extractable commitments, and that the later can be constructed based on any two-round semi-honest OT, e.g., [21], which implies PKE (see Sections 2.1 and 2.3 for more details), an immediate corollary from Theorem 32 implies that,

Corollary 33 *Assume the existence of two-round static semi-honest oblivious transfer. Then there exists a protocol that securely realizes* $\mathcal{F}_{\mathrm{OT}}$ *in the presence of static malicious adversaries in the CRS model using black-box access to the oblivious transfer protocol.*

A high level proof. We first provide an overview of the security proof; the complete proof is found in [30]. Loosely speaking, in case the receiver is corrupted the simulator plays the role of the honest sender in Stages 1-4. Next in Stage 5, the simulator extracts the receiver's input u. Specifically, the simulator extracts all the committed values of the receiver within Stage 1 (relying on the fact that the commitment scheme is extractable), and then uses these values in order to obtain the inputs for the OT executions in Stage 2. Upon completing Stage 2, the simulator records the coordinates for which the receiver deviates from the prescribed input and random tape chosen in the coin tossing phase. Denoting these set of coordinates by Φ, we recall that a malicious receiver may obtain both of the sender's inputs with respect to the OT executions that correspond to the coordinates within Φ and Γ. On the other hand, it obtains only one of the two inputs with respect to the rest of the OT executions that correspond to the coordinates within $\Delta - \Phi - \Gamma$. Consequently, the simulator checks how many shares of v_0 and v_1 are obtained by the receiver and proceeds accordingly. In more details,

- If the receiver obtains more than $10n$ shares of both inputs then the simulator halts and outputs fail (we prove in Section [30] that this event only occurs with negligible probability).
- If the receiver obtains less than $10n$ shares of both inputs then the simulator picks two random values for v_0 and v_1 of the appropriate length and completes the interaction, playing the role of the honest sender on these values. Note that in this case the simulator does not need to call the ideal functionality.
- Finally, if the receiver obtains more than $10n$ shares for only one input $u \in \{0, 1\}$, then the simulator sends u to the ideal functionality $\mathcal{F}_{\mathrm{OT}}$ and obtains v_u. The simulator then sets v_{1-u} as a random string of the appropriate length and completes the interaction by playing the role of the honest sender on these values.

Recall that the only difference between the simulation and the real execution is in the way the messages in Stage 4 are generated. Specifically, in the simulation a value u is extracted from the malicious receiver and then fed to the $\mathcal{F}_{\mathrm{OT}}$ functionality. The simulation is then completed based on the output returned from the functionality. Intuitively, the cut-and-choose mechanism ensures that the receiver cannot deviate from the honest strategy in Stage 2 in more than n OT sessions without getting caught with overwhelming probability. Moreover, the defensible privacy of the OT protocol implies that the receiver can learn at most one of the two inputs of the sender relative to the OT executions in Stage 2 for which the receiver proceeded honestly.

In case the sender is corrupted, the simulator's strategy is to play the role of the honest receiver until Stage 5 where the simulator extracts the sender's inputs. More specifically, the simulator first extracts the sender's input for the OT executions in Stage 1 (relying on the fact that the commitment scheme is extractable). Next, the simulator extracts the shares $\{\rho_i^0\}_{i\in\Delta}$ and $\{\rho_i^1\}_{i\in\Delta}$ that correspond to inputs v_0 and v_1. To obtain the actual values the simulator checks if these shares agree with some codeword relative to $16n$ locations. That is,

- Let w_0 and w_1 denote the corresponding codewords (if there are no such codewords that agree with with v_0 and v_1 on $16n$ locations then the simulator uses a default

codeword instead). Next, the simulator checks w_0 and w_1 against the final cut-and-choose. If any of the shares from w_b are inconsistent with the opened shares that are opened by the sender in the final cut-and-choose, then v_b is set to a default value, otherwise v_b is the value corresponding to the shared secret.

Finally, the simulator sends (v_0, v_1) to the ideal functionality for $\mathcal{F}_{\text{OT}}$. Security in this case is reduced to the privacy of the receiver. In addition, the difference between the simulation's strategy and the honest receiver's strategy is that the simulator extracts the sender's both inputs in all $i \in \Delta - \Phi$ and then finds codewords that are $16n$-close to the extracted values, whereas the honest receiver finds a codeword that is $17n$-close based on the inputs it received in the Stages 2 and 5, and returns it. We thus prove that the value u extracted by the simulator is identical the to the reconstructed output of the honest receiver relying on the properties of the secret sharing scheme.

4 One-Sided Adaptive UC Secure Computation

In the two-party one-sided adaptive setting, at most one of the parties is adaptively corrupted [35, 29]. In this section we provide a simple transformation of our static UC secure protocol from Section 3 to a two-party UC-secure protocol that is secure against one-sided adaptive corruption. Our first observation is that in Protocol 1 the parties use their real inputs to the OT protocol only in Phase 4. Therefore simulation of the first three phases can be easily carried out by simply following the honest strategy. On the other hand, simulating messages in Phase 4 requires some form of equivocation since if corruption takes place after this phase is concluded then the simulator needs to explain this message with respect to the real input of the corrupted party. On a high-level we will transform the protocol so that if no party is corrupted until end of Phase 4, the simulator can equivocate the message in Phase 4. We explain how to achieve equivocation later. First, we describe our simulator: In case either party is statically corrupted the simulation for Protocol 1 follows the strategy of the honest party until Phase 4, where the simulator extracts the corrupted party's input relying on the fact that it knows the adversary's committed input in Phase 1. Therefore, the same proof follows in case the adversary adaptively corrupts one of the parties at any point before Phase 4, as the simulator can pretend that corruption took place statically. On the other hand, if corruption takes place after Phase 4, then the simulator equivocates the communication. It is important to note that while in the plain model any statically secure protocol can be compiled into one-sided secure protocol by encrypting its entire communication, it is not clear that this is the case in the UC setting due to the additional setup, e.g., a CRS that may depend on the identity of the corrupted party. Nevertheless, in Phase 4 the parties only run a combiner for which the computation does not involve any usage of the CRS (which is induced by the extractable commitment). Therefore, the proof follows.

A common approach to achieve equivocation is to rely on non-committing encryption schemes (NCE) [6, 16, 11], that allow secure communication in the presence of adaptive attacks. This powerful tool has been constructed while relying on (a variant of) simulatable PKE schemes, which, roughly speaking, allows for both the public-key and the ciphertexts to be generated obliviously without the knowledge of the plaintext or the secret key [16, 11]. Notably, these constructions achieve a stronger notion of

security where both parties may be adaptively corrupted (also referred to as *fully adaptive*). Our second observation is that it is sufficient to rely on a weaker variant of NCE, namely, one that is secure against only one-sided adaptive corruption.

In particular, we take advantage of a construction presented in [6] and later refined in [16], that achieves receiver equivocation under the assumption of semi-honest OT. We will briefly describe it now. Recall that in the fully adaptive case, the high-level idea is for the sender and receiver to mutually agree on a random bit, which is then used by the sender to determine which of two random strings to mask its message. The process of agreeing on a bit requires the ability to both obliviously sample a public-key without the knowledge of the secret key, as well as the ability to obliviously sample a ciphertext without the knowledge of the corresponding plaintext. In the simpler one-sided scenario, Canetti et al. observed that an oblivious transfer protocol can replace the oblivious generation of the public-key. Specifically, the NCE receiver sends two public keys to the sender, and then the parties invoke an OT protocol where the NCE receiver plays the role of the OT sender and enters the corresponding secret keys. To allow equivocation for the NCE sender, the OT must enable equivocation with respect to the OT receiver. The [21] OT protocol is an example for such a protocol. Here the OT receiver can pick the two ciphertexts so that it knows both plaintexts. Then equivocation is carried out by declaring that the corresponding ciphertext is obliviously sampled.

The advantage of this approach is that it removes the requirement of generating the public key obliviously, as now the randomness for its generation is split between the parties, where anyway only one of them is corrupted. This implies that the simulator can equivocate the outcome of the protocol execution without letting the adversary the ability to verify it. To conclude, it is possible to strengthen the security of Protocol 1 into the one-sided setting by simply encrypting the communication within the combiner phase using one-sided NCE which in turn can be constructed based on PKE with oblivious ciphertext generation. This implies the following theorem which further implies black-box one-sided UC secure computation from enhanced trapdoor permutation.

Theorem 41 *Assume the existence of PKE with oblivious ciphertext generation. Then for any two-party well-formed functionality $\mathcal{F}$, there exists a protocol that UC realizes $\mathcal{F}$ in the presence of one-sided adaptive, malicious adversaries in the CRS model using black-box access to the PKE.*

5 Adaptive UC Secure Computation

In this section we demonstrate the feasibility of UC secure commitment schemes based on PKE with oblivious ciphertext generation (namely, where it is possible to obliviously sample the ciphertext without knowing the plaintext). Our construction is secure even in the presence of adaptive corruptions and is the first to achieve the stronger notion of adaptive security based on this hardness assumption. Plugging-in our UC commitment protocol into the transformation of [12] that generates adaptive malicious OT given adaptive semi-honest OT and UC commitments, implies an adaptively UC secure oblivious transfer protocol with malicious security based on semi-honest adaptive OT and PKE with oblivious ciphertext generation using only black-box access to the semi-honest OT and the PKE. Stating formally,

Theorem 51 *Assume the existence of adaptive semi-honest oblivious transfer and PKE with oblivious ciphertext generation. Then for any multi-party well-formed functionality $\mathcal{F}$, there exists a protocol that UC realizes $\mathcal{F}$ in the presence of adaptive, malicious adversaries in the CRS model using black-box access to the oblivious transfer protocol and the PKE.*

Noting that simulatable PKE implies both semi-honest adaptive OT [9, 11] and PKE with oblivious ciphertext generation, we derive the following corollary (where simulatable PKE implies oblivious sampling of both public keys and ciphertexts),

Corollary 52 *Assume the existence of simulatable PKE. Then for any multi-party well-formed functionality $\mathcal{F}$, there exists a protocol that UC realizes $\mathcal{F}$ in the presence of adaptive, malicious adversaries in the CRS model using black-box access to the simulatable PKE.*

This in particular improves the result from [14] that relies on simulatable PKE in a non-black-box manner. Note also that our UC commitment can be constructed using a weaker notion than simulatable PKE where the inverting algorithms can require a trapdoor. This notion is denoted by trapdoor simulatable PKE [11] and can be additionally realized based on the hardness assumption of factoring Blum integers. This assumption, however, requires that we modify our commitment scheme so that the CRS includes $3n + 1$ public keys of the underlying PKE instead of just one, as otherwise the reduction to the security of the PKE does not follow for multiple ciphertexts. Specifically, at the cost of linear blowup (in the security parameter) of the CRS, we obtain adaptively secure UC commitments under a weaker assumption. Now, since trapdoor simulatable PKE implies adaptive semi-honest OT [11] it holds,

Corollary 53 *Assume the existence of trapdoor simulatable PKE. Then for any multi-party well-formed functionality $\mathcal{F}$, there exists a protocol that UC realizes $\mathcal{F}$ in the presence of adaptive, malicious adversaries in the CRS model using black-box access to the trapdoor simulatable PKE.*

Note that, since the best known general assumptions for realizing adaptive semi-honest OT is trapdoor simulatable PKE, this corollary gives evidence that the assumptions for adaptive semi-honest OT are sufficient for adaptive UC security and makes a step towards identifying the minimal assumptions for achieving UC security in the adaptive setting. To conclude, we note that enhanced trapdoor permutations, which imply PKE with oblivious ciphertext generation, imply the following corollary,

Theorem 54 *Assume the existence of enhanced trapdoor permutation. Then $\mathcal{F}_{\mathrm{COM}}$ (cf. Figure 2) can be UC realized in the CRS model in the presence of adaptive malicious adversaries.*

5.1 UC Commitments from PKE with Oblivious Ciphertext Generation

In this section we demonstrate the feasibility of adaptively secure UC commitments for the message space $m \in \{0,1\}$ from any public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Enc}}^{-1})$ with oblivious ciphertext generation (cf. Definition 21)

in the common reference string (CRS) model. In this model [7] the parties have access to a CRS chosen from a specified trusted distribution $\mathcal{D}$. This is captured via the ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$ (see [30] for the definition). We note that we use Π in two places in our protocol. First, in the encoding phase (where the commitments are computed by the sender) and then in the coin-tossing phase (where the commitments are computed by the receiver). Our complete construction can be found in Figure 4. Next, we prove

Protocol π_{COM}.

CRS: Two independent keys $\mathrm{PK}, \widetilde{\mathrm{PK}}$ that are in the range of $\mathsf{Gen}(1^n)$.

Sender's Input: A message $m \in \{0,1\}$ and a security parameter 1^n.

[Commitment phase:]

Encoding phase: The sender chooses a random n-degree polynomial $p(\cdot)$ over a field $\mathbb{F}[x]$ such that $p(0) = m$. Namely, it randomly chooses $a_i \leftarrow \mathbb{F}$ for all $i \in [n]$ and sets $a_0 = m$, and defines the polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. The sender then creates a commitment to m as follows. For every $i = [3n+1]$, it first pick $b_i \leftarrow \{0,1\}$ at random and then computes the following pairs:

$$\text{If } b_i = 0 \text{ then } \begin{array}{l} c_i^0 = \mathsf{Enc}_{\mathsf{PK}}(p(i); t_i) \\ c_i^1 = r_i \end{array} \quad \text{else,} \quad \text{if } b_i = 1 \text{ then } \begin{array}{l} c_i^0 = r_i \\ c_i^1 = \mathsf{Enc}_{\mathsf{PK}}(p(i); t_i) \end{array}$$

where $t_i \leftarrow \{0,1\}^n$ and $r_i \leftarrow \widetilde{\mathsf{Enc}}(\cdot)$ is obliviously sampled. The sender sends $(c_0^0, c_0^1), \ldots, (c_{3n+1}^0, c_{3n+1}^1)$ to the receiver.

Coin-tossing phase: The sender and receiver interact in a coin-tossing protocol that is carried out as follows.

1. The receiver sends $c = \mathsf{Enc}_{\widetilde{\mathrm{PK}}}(\sigma_0; r_{\sigma_0})$ to the sender where $\sigma_0 \leftarrow \{0,1\}^N$ is chosen uniformly at random.
2. The sender picks $\sigma_1 \leftarrow \{0,1\}^N$ at random and sends it in the clear to the receiver
3. The receiver decrypts c by revealing σ_0 and r_{σ_0}.

Both the sender and the receiver compute $\sigma = \sigma_0 \oplus \sigma_1$ and use σ as the random string to sample a random subset $S \subset [3n+1]$ of size n. (Note that such sampling can be done in a simple way by partitioning the set of coordinates into n sets of triples (where the last set includes 4 elements) and picking one element per set. Notably, this technique does not imply that any potential subset of size n will be picked, rather it ensures that a subset is picked with a negligible probability in n, specifically $(1/3)^n$, which suffices for our proof.)

Cut-and-choose phase: The sender decrypts the set $\{c_i^{b_i}\}_{i \in S}$ by sending the sequence $\{b_i, p(i), t_i\}_{i \in S}$. The receiver verifies that all the decryptions are correct and aborts otherwise.

[Decommitment phase:] Let $T = [3n+1] - S$. The sender reveals its input m and decrypts all the ciphertexts in $\{c_i^{b_i}\}_{i \in T}$. The receiver checks if all the decryptions are correct and aborts otherwise. Using the n polynomial evaluations revealed relative to $i \in S$ and any additional polynomial evaluation that was revealed relative to T, the receiver reconstructs the polynomial $p(\cdot)$ (via polynomial interpolation of $n+1$ points). Next, the receiver verifies whether $p(0) = m$, and that for every $i \in [3n+1]$ the point $p(i)$ is the decrypted value within $c_i^{m_i}$.

Fig. 4. UC adaptively secure commitment scheme.

Theorem 55 *Assume that* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Enc}}^{-1})$ *is a PKE with oblivious ciphertext generation. Then protocol* π_{COM} *(cf. Figure 4) UC realizes* $\mathcal{F}_{\mathrm{COM}}$ *in the CRS model in the presence of adaptive malicious adversaries.*

A high level proof. Intuitively, security requires proving both hiding and binding in the presence of static and adaptive corruptions. The hiding property follows from the IND-CPA security of the encryption scheme combined with the fact that the receiver only sees n shares in a n-out-of-$3n + 1$ secret-sharing of the message in the commit phase. On the other hand, proving binding is much more challenging and reduces to the facts that a corrupted sender cannot successfully predict exactly the n indices from $\{1, \ldots, 3n + 1\}$ that will be chosen in the coin-tossing protocol. In fact, if it can identify these n indices, then it would be possible for the adversary to break binding. An important information-theoretic argument that we prove here is that for a fixed encoding phase, no adversary can equivocate on two continuations from the encoding phase with different outcomes of the coin-tossing phase. Saying differently, for any given encoding phase there is exactly one outcome for the coin-tossing phase that will allow equivocation. Given this claim, binding now follows from the IND-CPA security of the encryption scheme used in the coin-tossing phase. In addition, recall that in the UC setting the scheme must also support a simulation that allows straight-line extraction and equivocation. At a high-level, the simulator sets the CRS to public-keys for which it knows the corresponding secret-keys. This will allow the simulator to extract all the values encrypted by the adversary. We observe that the simulator can fix the outcome of the coin-tossing phase to any n-indices of its choice by extracting the random string σ_0 encrypted by the receiver and choosing a random string σ_1 so that $\sigma_0 \oplus \sigma_1$ is a particular string. Next, the simulator generates secret-sharing for both 0 and 1 so that they overlap in the particular n shares. To commit, the simulator encrypts the n common shares within the n indices to be revealed (which it knows in advance), and for the rest of the indices it encrypts two shares, one that corresponds to the sharing of 0 and the other that corresponds to the sharing of 1. Finally, in the decommit phase, the simulator reveals that shares that correspond to the real message m, and exploits the invertible sampling algorithm to prove that the other ciphertexts were obliviously generated.

References

1. Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS*, pages 186–195, 2004.
2. Donald Beaver. Foundations of secure interactive computing. In *CRYPTO*, pages 377–391, 1991.
3. Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. Analysis and improvement of lindell's uc-secure commitment schemes. In *ACNS*, pages 534–551, 2013.
4. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.
5. Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. *IACR Cryptology ePrint Archive*, 2006:432, 2006.
6. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.

7. Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO*, pages 19–40, 2001.
8. Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. *J. Cryptology*, 19(2):135–167, 2006.
9. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.
10. Ran Canetti, Rafael Pass, and Abhi Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *FOCS*, pages 249–259, 2007.
11. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *ASIACRYPT*, pages 287–302, 2009.
12. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In *TCC*, pages 387–402, 2009.
13. Seung Geol Choi, Jonathan Katz, Hoeteck Wee, and Hong-Sheng Zhou. Efficient, adaptively secure, and composable oblivious transfer with a single, global CRS. In *PKC*, pages 73–88, 2013.
14. Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Muthuramakrishnan Venkitasubramaniam. Adaptive and concurrent secure computation from new adaptive, non-malleable commitments. In *ASIACRYPT*, pages 316–336, 2013.
15. Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC*, pages 426–437, 2003.
16. Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO*, pages 432–450, 2000.
17. Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002.
18. Ivan Damgård, Jesper Buus Nielsen, and Claudio Orlandi. On the necessary and sufficient assumptions for UC computation. In *TCC*, pages 109–127, 2010.
19. Bernardo David, Rafael Dowsley, and Anderson C. A. Nascimento. Universally composable oblivious transfer based on a variant of LPN. In *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, pages 143–158, 2014.
20. Bernardo Machado David, Anderson C. A. Nascimento, and Jörn Müller-Quade. Universally composable oblivious transfer from lossy encryption and the mceliece assumptions. In *Information Theoretic Security - 6th International Conference, ICITS 2012, Montreal, QC, Canada, August 15-17, 2012. Proceedings*, pages 80–99, 2012.
21. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.
22. Juan A. Garay, Daniel Wichs, and Hong-Sheng Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In *CRYPTO*, pages 505–523, 2009.
23. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335, 2000.
24. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.
25. Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *FOCS*, pages 51–60, 2012.
26. Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In *TCC*, pages 412–426, 2008.

27. Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011.
28. Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1):158–193, 2012.
29. Carmit Hazay and Arpita Patra. One-sided adaptively secure two-party computation. In *TCC*, pages 368–393, 2014.
30. Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On black-box complexity of universally composable security in the CRS model. *IACR Cryptology ePrint Archive*, 2015:488, 2015.
31. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *CRYPTO*, pages 8–26, 1988.
32. Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In *STOC*, pages 99–108, 2006.
33. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO*, pages 572–591, 2008.
34. Yael Tauman Kalai, Yehuda Lindell, and Manoj Prabhakaran. Concurrent composition of secure protocols in the timing model. *J. Cryptology*, 20(4):431–492, 2007.
35. Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *CRYPTO*, pages 335–354, 2004.
36. Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31, 1988.
37. Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In *CRYPTO*, pages 461–478, 2012.
38. Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In *STOC*, pages 179–188, 2009.
39. Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for UC from only OT. In *ASIACRYPT*, pages 699–717, 2012.
40. Yehuda Lindell. General composition and universal composability in secure multi-party computation. In *FOCS*, pages 394–403, 2003.
41. Yehuda Lindell. Adaptively secure two-party computation with erasures. In *CT-RSA*, pages 117–132, 2009.
42. Yehuda Lindell. Highly-efficient universally-composable commitments based on the DDH assumption. In *EUROCRYPT*, pages 446–466, 2011.
43. Yehuda Lindell and Hila Zarosim. Adaptive zero-knowledge proofs and adaptively secure oblivious transfer. In *TCC*, pages 183–201, 2009.
44. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A zero-one law for cryptographic complexity with respect to computational UC security. In *CRYPTO*, pages 595–612, 2010.
45. Silvio Micali and Phillip Rogaway. Secure computation (abstract). In *CRYPTO*, pages 392–404, 1991.
46. Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 448–457, 2001.
47. Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.
48. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.
49. M. Rabin. How to exchange secrets by oblivious transfer. Tech. Memo TR-81, Aiken Computation Laboratory, Harvard U., 1981.
50. Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FCOS*, pages 162–167, 1986.