A Unified Framework for UC from Only OT

Huijia Lin[1], Rafael Pass[2], and Muthuramakrishnan Venkitasubramaniam[3]

[1] MIT and Boston University, Boston, MA, 02138, USA
[2] Cornell University, Ithaca NY 14850, USA
[3] University of Rochester, Rochester, NY 14611, USA

Abstract. In [1], the authors presented a unified framework for constructing Universally Composable (UC) secure computation protocols, assuming only enhanced trapdoor permutations. In this work, we weaken the hardness assumption underlying the unified framework to only the existence of a stand-alone secure semi-honest Oblivious Transfer (OT) protocol. The new framwork directly implies new and improved UC feasibility results from only the existence of a semi-honest OT protocol in various models. Since in many models, the existence of UC-OT implies the existence of a semi-honest OT protocol.

Furthermore, we show that by relying on a more fine-grained analysis of the unified framework, we obtain concurrently secure computation protocols with super-polynomial-time simulation (SPS), based on the *necessary* assumption of the existence of a semi-honest OT protocol that can be simulated in super-polynomial times. When the underlying OT protocol has constant rounds, the SPS secure protocols constructed also have constant rounds. This yields the first construction of constant-round secure computation protocols that satisfy a meaningful notions of concurrent security (i.e., SPS security) based on tight assumptions.

A notable corollary following from our new unifed framwork is that stand-alone (or bounded-concurrent) password authenticated key-exchange (PAKE) protocols can be constructed from only semi-honest OT protocols; combined with the result of [2] that the existence of PAKE protocols implies that of OT, we derive a tight characterization of PAKE protocols.

1 Introduction

The notion of *secure multi-party computation* allows m mutually distrustful parties to securely compute a functionality $f(\bar{x}) = (f_1(\bar{x}), ..., f_m(\bar{x}))$ of their corresponding private inputs $\bar{x} = x_1, ..., x_m$, such that party P_i receives the value $f_i(\bar{x})$. Loosely speaking, the security requirements are that the parties learn nothing more from the protocol than their prescribed output, and that the output of each party is distributed according to the prescribed functionality. This should hold even in the case that an arbitrary subset of the parties maliciously deviates from the protocol.

Shortly after the notion was proposed, strong results were established for secure multi-party computation. Specifically, it was shown that any probabilistic polynomial-time computable multi-party functionality can be securely computed, assuming existence of enhanced trapdoor permutations [3, 4]. The original

setting in which secure multi-party protocols were investigated, however, only allowed the execution of a single instance of the protocol at a time; this is the so called *stand-alone setting*. A more realistic setting, is one which allows the concurrent execution of protocols. In the *concurrent setting*, many protocols are executed at the same time. This setting presents the new risk of a coordinated attack in which an adversary interleaves many different executions of a protocol and chooses its messages in each instance based on other partial executions of the protocol. The strongest (but also most realistic) setting for concurrent security—called *Universally Composable* (UC) security [5]—considers the execution of an unbounded number of concurrent protocols, in an arbitrary, and adversarially controlled, network environment. Unfortunately, security in the stand-alone setting does not imply security in the concurrent setting. In fact, without assuming some trusted set-up, the traditional simulation-based notion of concurrent security, and in particular UC security, cannot be achieved in general [6–8].

To circumvent the broad impossibility results, two distinct veins of research can be identified in the literature.

Trusted set-up models: A first vein of work initiated by Canetti and Fischlin [6] and Canetti, Lindell, Ostrovsky and Sahai [9] (see also e.g., [10–13]) considers constructions of UC-secure protocol using various trusted set-up assumptions, where the parties have limited access to a trusted entity.

Relaxed models of security: Another vein of work considers relaxed models of security such as *quasi-polynomial simulation* [14–16] or *input-indistinguishability* [17]. These works circumvents the use of trusted set-ups, but, only provide weak guarantees about the computational advantages gained by an adversary in a concurrent execution of the protocol.

In [1], we provided a general unified framework to construct UC-secure protocols in both trusted set-up models and relaxed security models. In more detail, we showed that for any such model, the construction of UC protocols for realizing any multi-party functionality reduces to the construction of a so-called "UC-puzzle" and a so-called strongly non-malleable witness indistinguishable ($\mathcal{SNMWI}$) argument of knowledge. Intuitively, a "UC-puzzle" is a protocol that has the property that no adversary can successfully complete the puzzle and also obtain a trapdoor, but there exists a simulator who can generate (correctly distributed) puzzles together with trapdoors; and a $\mathcal{SNMWI}$ argument ensures that no man-in-the-middle adversary can correlate the witness it uses in a proof with the witness in the proof it receives[4]. They we showed that a $\mathcal{SNMWI}$ argument can be implemented using any non-malleable commitment scheme; therefore the task of realizing UC security in any model reduces to the task of constructing a "UC-puzzle" in that model, which can be easily achieved in almost all previously considered set-up and relaxed security models. Furthermore, in many models, we showed that the existence of a "UC-puzzle" is also

[4] A $\mathcal{SNMWI}$ argument can be viewed as an analogy of non-malleable commitments in the context of strongly $\mathcal{WI}$ proofs [18].

necessary; in a sense, the notion of "UC-puzzle" characterizes the "minimal" set-up and relaxation of security needed for achieving UC security.

In this work, we focus on a different dimension: Namely, given the minimal set-up and relaxation of security need for UC, what is the "minimal" computational assumption additionally needed for constructing UC secure protocols. In [1], the construction of UC protocols from "UC-puzzles" is based on the existence of enhanced trapdoor-permutations (TDP's), whereas stand-alone secure multi-party computation protocols can be constructed based on the minimal assumption of the existence of stand-alone secure semi-honest OT protocols [19, 20], which clearly also is a necessary assumption. This immediately raises the following question.

Can we base UC security on the minimal assumption of the existence of a semi-honest OT protocol?

1.1 Previous Works

Immediately after the work of [1], there has been several works trying to address this problem in specific models.

In KRA and CRS model: Damgard, et al. [21] showed that UC security can be achieved assuming only semi-honest OT protocols in the key registration (KR), and common reference string (CRS), as well as uniform reference string (URS) models. Their constructions in the KR, and the more generalized arbitrary KR (A-KR), models achieve optimal round complexity, which have the same number of rounds as the underlying semi-hoest OT protocol (up to a constant factor). However, the round-complexity of their construction in the CRS and URS model grows linearly with the number of players in the protocol execution. Furthermore, their construction in the CRS and URS model only implements an ideal functionality $\mathcal{F}$ in a *single session*, meaning every execution of their protocol needs to invoke the CRS functionality to obtain an independently sampled reference string. In contrast, previous constructions of UC secure protocol in the CRS model directly implements the *multi session extension of* $\mathcal{F}$ [9, 1] so that different protocol executions may share the same CRS.

In the $\mathcal{F}_{\mathsf{coin-toss}}$ hybrid model: In the context of characterizing functionalities that are complete for achiveing UC security, Maji, Prabhakaran and Rosulek [22] showed that the *ideal two-party coin-tossing functionality* $F_{\mathsf{coin-toss}}$ is "complete", in the sense that, assuming the existence of semi-honest OT protocols, practically all functionalities[5] can be UC-securely realized when players have access to the $\mathcal{F}_{\mathsf{coin-toss}}$ functionality, with the same round complexity as the OT protocol.

In the tamper-proof hardware model: Goyal et. al. showed that in the model where players can generate and exchange tamper-proof hardware tokens, UC

[5] More precisely, all well-formed functionalities can be UC-securely realized.

security can be achived assuming the weaker assumption of one-way functions or even unconditionally, in a constant-number of rounds.

The above mentioned previous works try to weaken the assumptions that UC security is based on using different techniques and exploiting different features of the specific models under consideration. This immediately raises the question whether we can achieve UC security from semi-honest OT protocol in a generic way as in [1], independent of the specifics of different set-up or relaxed security models.

> *Can we base UC security only on the existence of semi-honest OT protocols,* generically*?*
> *Furthermore, can we achieve so with* optimal round complexity*?*

Such a generic construction would not only help us identify and undersand the key elements needed for achieving UC security, also allow us to obtain new UC-feasibility results in other models easily.

Furthermore, one common limitation of the previous results is that they all used the trusted set-ups in a strong way so that different protocol executions have *different and independent* "trapdoors", which makes UC security relatively easy to achieve. Let us explain the intuition. In order to construct a protocol secure in the concurrent setting, we need to establish two properties: *Concurrent simulation*, that is, the simulator can simulate messages from the honest players in many concurrent sessions for the adversary, and *concurrent simulation-soundness*, that is the adversary even when receiving simulated messages cannot break the security of the protocol against honest players. The concurrent simulation property can be established easily as long as there is a single trapdoor (or correlated trapdoors) shared by all protocol executions; the simulator can simply use that trapdoor to simulate. The concurrent simulation-soundness property, on the other hand, is much harder to establish, and often involves the use of *non-malleable* primitives to ensure independence of different protocol executions as in [9, 1, 23]. However, in the case where different sessions have independent trapdoors, concurrent simulation-soundness can be obtained "for free", as receiving simulated messages (containing information of one trapdoor) does not help the adversary obtain other trapdoors; hence, the security of the protocol w.r.t. the honest players remains.

Indeed, all previous works use the trusted set-up to generate independent trapdoors for different protocol executions. In the CRS (resp. URS) model, [21] constructed protocols that implement a general functionality $\mathcal{F}$ in a *single session*, meaning that each executions of their protocol invokes the CRS (resp. URS) functionality independently, which yields independent trapdoors (that is, independent secrets associated with different CRS's (resp. URS's)). In the KR and A-KR model of [21], every player is registered with a valid public key that has a corresponding secret key; furthermore, the secret key of any honest player is hidden even if the adversary obtains the secret keys of all other players. Naturally, the secret keys of players are used as independent trapdoors. The same happens in the tamper-proof hardware token model, where the freshly generated

hardware tokens in each session yield independent trapdoors for different session. Finally, in the ideal coin-tossing hybrid model, the $F_{\mathsf{coin}-\mathsf{toss}}$ functionality is used to sample an independent URS in every session.

However, in many "weaker" models, there is only a single trapdoor (or correlated trapdoors) across many protocol executions. Then the techniques used in previous works no longer apply, and the protocol construction needs to explicitly "inject" independence to establish simulation soundness. Such set-up models include the CRS model, when the protocol construction directly implements the multi-session extension of functionalities, the single imperfect string (sunspot) model [13], the timing model [24] and the bounded concurrency model [25]. Furthermore, the super-polynomial time simultion model also share the same flavor: Though each protocol execution session may generate its own trapdoor (for instance, the pre-image of a randomly sampled image of a one-way function), receiving information of the trapdoor in one session, obtained via the super-polynomial time power of the simulator, does facilitate the adversary breaking the trapdoor in other sessions, as the adversary may create correlation between trapdoors in different sessions. Naturally, the question left open by previous works is,

> *Can we construct UC secure protocols when there are only* correlated trapdoors*, based on the existence of semi-honest OT protocols?*

1.2 Our Results

In this work, we answer both questions above affirmatively. We improve upon the result in [1] to obtain a new unified framework for constructing UC secure protocols, assuming only the existence of semi-honest OT protocols. More precisely, the main theorem that we establish is:

Theorem 1 (Unified Framework from OT, Informal). *Assume the existence of a $t_1(\cdot)$-round UC-secure puzzle Σ using some set-up $\mathcal{T}$, and the existence of a $t_2(\cdot)$-round stand-alone secure semi-honest oblivious-transfer protocol. Then, for every m-ary functionality f, there exists a $O(t_1(\cdot) + t_2(\cdot))$-round protocol Π—using the same set-up $\mathcal{T}$—that UC-realizes the multi-session extension of f.*

We remark that since our main theorm is general and only requires the security model to admit a single UC-puzzle, the unified framework we provide encompasses both models where there are only correlated trapdoors, as detailed below.

Trusted Set-up Models: As shown in [1], many trusted set-up models admit *constant-round* UC-puzzles assuming the existence of one-way functions. Thus, our unified framework immediately yields UC feasibility results from only semi-honest OT, in a wide range of set-up models.

Corollary 1 (Trusted Set-up Models). *Assume the existence of a $t(\cdot)$-round stand-alone secure semi-honest oblivious-transfer protocol. Then, for every m-ary functionality f, there exists a $O(t(\cdot))$-round protocol Π that UC-realizes the multi-session extension of f in the following models:*

- *Tamper proof hardware model [26],*
- *Key registration (KR) model [10]*
- *Chosen common reference string (C-CRS) model [9], any common reference string (A-CRS) model [21], and uniform reference string (URS) model [9],*
- *Timing model [24],*
- *Multi-string model [27],*
- *Single imperfect string (sun-spot) model [13] (assuming additionally the existence of collision resistence hash functions).*

We compare our results with previous works. In the tamper-proof hardware model (line 1), our feasibility result is weaker than that of [28], which achieved UC unconditionally. In the key-registration models (line 2), we re-prove the result in [21]. In the CRS and URS models (line 3), we obtain new feasibility results that implement directly the multi-session extension of functionalities, instead of implementing only in single session as in [21]; furthermore, we improve the round complexity to that of the OT protocol, whereas in [21] the round-complexity grows linearly with the number of players in the protoccol execution. In the rest of set-up models (line 4 to 6) that only admit correlated trapdoors, we obtain new UC feasilibity results from only semi-honest OT.

Optimal Round-Complexity: We remark that round-complexity of our construction depends solely on and is at the same order as that of the underlying semi-honest OT protocol. Therefore, assuming the existence of a *constant-round* semi-honest OT protocol, we obtain *constant-round* UC secure protocols in all above mentioned models.

Sufficient and Necessary Assumption for UC Security: Our main theorem shows that t-round semi-honest OT protocols are sufficient for UC security in various models. In fact, it is also necessary in many models. As shown in [29, 21], t-round UC secure computation in the key registration, CRS and URS models (line 1 and 2) implies t-round semi-honest OT; since the single-CRS, and single-URS models are strictly weaker than their one-CRS-per-session and one-URS-per-session versions, the implication also holds in these two models. It is easy to see that the same is true in the timing model. Therefore, our result yields a *tight* characterization of the feasibility of t-round UC secure computation (from $\Omega(t)$-round semi-honest OT) in the key-registration, CRS, URS, single-CRS, single-URS and timing models.

Super-Polynomial Time Simulation Model In a super-polynomial time simulation model with simulation time T—T can be, say, quasi-polynomial time (QPT) or sub-exponential time (subEXP)—assuming the existence of a one-way function that is hard to invert in polynomial time, but easy to invert (with probability 1) in T time, there exists a one-message UC-puzzle in T-time simulation model[6]. Note that when considering subEXP time simulation, the assumption of

[6] The UC puzzle simply consists of one message from the sender is the image of a random string through that one-way function. It is hard for polynomial time adversary to break the puzzle (i.e., obtain a pre-image), but easy for a T-time simulator.

one-way functions invertable in subEXP time is simply implied by the existence of any one-way functions[7]. Therefore, applying our main theorem[8], we have:

Corollary 2 (Super-Polynomial Time Simulation Models). *Assume the existence of a $t(\cdot)$-round stand-alone secure semi-honest oblivious-transfer protocols* secure for subEXP-time*. Then, for every m-ary functionality f, there exists a $O(t(\cdot))$-round protocol Π that realizes f with* subEXP-time-simulation security*. Furthermore, the real and ideal executions are indistinguishable to all* subEXP-time distinguishers*.*

This result weakens the assumptions that SPS secure protocols can be relied on: Previous constructions either requires strong complexity assumptions [15, 16] or the existence of enhanced trapdoor permutations secure against super-polynomial time [1].

Moreover, Our subEXP-secure protocols have optimal round-complexity. The construction relies on the existence of semi-honest OT protocols that are secure for subEXP time (i.e., semi-honest OT protocol that are simulatable by subEXP-time simulator and the simulation is indistinguishable to the real execution to subEXP-time distinguishers). This assumption is in fact *necessary*, in order to achieve the strong security guarantees provided by our unified framework: Protocols constructed through our unified framework admits simulation (i.e., the ideal world execution) that are indistinguishable from the real execution not only to all polynomial time distinguishers, but also to distinguishers with the same running time as the simulator; we call this *strong SPS-security*.

Constant-Round SPS Security from Poly-Time Secure OT. As discussed above, strong SPS security necessarily relies on super-polynomial time hard OT protocol. We show that, in fact, the use of super-polynomial time hardness assumption can be circumvented, when considering a weaker notion of security called *plain SPS-security*, where the simulator may take super-polynomial time, but the simulation produced is only indistinguishable w.r.t. polynomial time. (In fact, this is the security guarantee achieved in the first two positive results of SPS security in [15, 16], although they still requried super-polynomial time hardness assumptions.) Given a semi-honest OT protocol that is simulatable in subEXP-time but only indistinguishable to $\mathcal{PPT}$ distinguishers—call it a subEXP-simulatable semi-honest OT protocol—we have,

Theorem 2 (Plain SPS-security from Polynomial-time OT). *Assume the existence of a $t(\cdot)$-round stand-alone secure subEXP-simulatable semi-honest*

[7] Every one-way function can be inverted in exponential time using brute force. Therefore, by appropriately scale down the security parameter, we obtain one-way functions that can be inverted in sub-exponential time.

[8] The informal statement of our unified framework in Theorem 1 does not explicitly specify the complexity of the simulator and distinguisher, nor their relationship with the hardness of the OT in the assumption. More precisely, our unified framework holds for arbitrary classes $\mathcal{C}_{\mathsf{sim}}$ of simulators and distinguishers, assuming an OT protocol that is secure for $\mathcal{C}_{\mathsf{sim}}$. See Section 3 for a formal treatment of the security definition and statement of our unified framework in Theorem 3.

oblivious-transfer protocol. Then, for every m-ary functionality f, there exists a $O(t(\cdot))$-round protocol Π that realizes f with plain subEXP-time-simulation security.

Recently, Canetti, Lin, and Pass in [30] showed how to achieve plain SPS-security, assuming only enhanced trapdoor permutations; however, their construction requires polynomially many communication rounds, whereas our construction yields constant-round protocols assuming that the underlying OT protocol has constant rounds. In concurrent and independent work, Garg, Goyal, Jain and Sahai [31], also present a construction of constant-round SPS secure protocols; but they additionally assume the existence of collision resistant hash functions besides from that of semi-honest OT.[9] Finally, we remark that our assumption is again tight: secure protocols with plain subEXP-time-simulation security imply OT protocols that can be simulated using subEXP time.

Password-Key Exchange from OT As another application of our unified framework, we consider another line of relaxation—bounded concurrency—that is, in the concurrent execution of protocols, there is *a priori* bound on the total number of sessions that may coexist at any time point. This line of relaxation has been previously considered in several works [32, 33, 8, 25]; they showed how to construct bounded-concurrent secure computation using non black-box techniques, based on the existence of collision resistant hash functions. We show that in fact, the model of bounded concurrency can be cast as a special case of our generalized model of UC security, by considering a restricted class of environment that respects the bound m_2 on the total number of concurrent executions, and additionally only exchanges a bounded number m_1 of messages with the the adversary. We call this the (m_1, m_2)-bounded concurrency model. Therefore, by constructing a $O(m_1 + m_2)$ UC-puzzle in this model, we immediately obtain the following feasibility result.

Corollary 3 (Bounded Concurrency Model). *Let m and m' be any polynomial. Assume the existence of* constant-round *stand-alone secure semi-honest oblivious-transfer protocol. Then, for every m-ary functionality f, there exists a $O(m_1 + m_2)$-round protocol Π that securely realizes f in the (m_1, m_2)-bounded concurrency model.*

Lindell [34] showed that $O(m)$ communication rounds are necessary for security in the $(m, 0)$-bounded concurrency model, when relying on black-box simulation techniques; therefore, our construction achieves the optimal round-complexity. Furthermore, it is shown in [**?**] that the existence of t-round two-party computation protocols in the $(2, 0)$-bounded concurrency model implies the existence of t Password-Authenticated Key-Exchange (PAKE) protocols. Therefore, we obtain $O(t)$-round PAKE protocols from any t-round semi-honest OT. Combined with the result of Nguyen [2] that t-round PAKE implies $O(t)$-round OT, this

[9] Their proof techniques, however, are significantly different, and it would seem that an advantage of their approach is that they not rely on non-uniform reductions to an as large extent as we do.

resolves the complexity of PAKE protocols. Previous constructions of PAKE protocols assume stronger assumptions, namely, the existence of enhanced trapdoor permutations and collision resistant hash functions [**?**]. Another related work due to Goyal, Jain and Ostrovsky [35] considered a weaker notion of security[10], and constructed PAKE protocols satisfying the weaker notion in the unbounded concurrent setting based on collision resistant hash functions.

1.3 Outline

We refer the reader to [1] for a formal definition of the generalized model of UC-security, and notions of UC-puzzle and $\mathcal{SNMWI}$ protocols. In Section 2 provide an overview of our techniques. In Section 3, we present our main result that general UC security can be based on sh-OT protocols, and provide a proof sketch. We defer the formal description of the rest of our results and all formal proofs to the full version.

2 Techniques

2.1 The LPV Approach

By relying on previous results [25, 33, 36, 9, 4] the construction of a UC secure protocol for realizing any multi-party functionality reduces to the task of realizing the "ideal Zero-Knowledge functionality", which amounts to constructing a zero-knowledge protocol that is both *concurrently simulatable* and *concurrently simulation-extractable*—namely, we can concurrently extract a witness from every convincing proof given by the adversary, even if it receives multiple concurrent *simulated* proofs. The "simulation" part is usually easy to achieve; as shown in [1], it suffices to provide the simulator a single "trapdoor". This is formalized by the notion of a UC-puzzle in [1], which, intuitively, is a protocol that has the property that no adversary can successfully complete the puzzle and also obtain a trapdoor, but there is a simulator who can generate puzzle transcripts (distributed statistically close to real transcripts) together with trapdoors; the former is called the *soundness* property and the latter called the *statistical simulation* property. However, obtaining "simulation-soundness" it significantly harder. In [1], the authors achieve this in two steps: First construct a "special-purpose" zero-knowledge protocol that is *concurrently simulation-sound*—namely, even if an adversary receives multiple concurrent simulated proofs, it will not be able to prove any false statements; then, enhance the security to get simulation-extractability.

[10] More precisely, the security notion of [35] is defined through the simulation paradigm where the simulator may rewind the trusted functionality, for instance, the ideal PAKE functionality, for a limited number of times, whereas we achieve full security without rewinding. On the other hand, their protocols are secure in unbounded concurrent setting, however, ours are only secure in bounded concurrent setting

The first step relies on a primitive called strong non-malleable witness-indistinguishable ($\mathcal{SNMWI}$) arguments, which captures the *non-malleability* property w.r.t. strongly witness indistinguishable proofs. Informally, a $\mathcal{SNMWI}$ argument ensures that no man-in-the-middle adversary can correlate the witness it uses in a proof with the witness in the proof it receives. It is shown in [1] that $\mathcal{SNMWI}$ arguments can be constructed from non-malleable commitments. At a high-level, the simulation-sound protocol follows the Feige-Shamir paradigm, in which the verifier first sends a UC-puzzle to establish a "trapdoor" (that is, the puzzle answer), and then the prover proves that either the statement is true or it knows a trapdoor, using a $\mathcal{SNMWI}$ argument[11]. In essence, the UC-puzzle enables concurrent simulation: A simulator can simulate the puzzle executions with the verifier to obtain corresponding answers, and then use them as trapdoors to successfully simulate the $\mathcal{SNMWI}$ arguments. On the other hand, the $\mathcal{SNMWI}$ property ensures simulation-soundness: Even if the adversary receives $\mathcal{SNMWI}$ proofs using the trapdoors as "fake witnesses", the adversary does not do the same.

The second step in [1] enhances the security by employing the compilation technique of [**?**,36, 33], which transforms a concurrently simulation-sound protocol into one that is concurrently simulation-extractable, using enhanced trapdoor permutations (TDP).

2.2 UC-Security from Semi-Honest OT

In this work, we weaken the assumption that UC security relies on, by providing a new compilation technique for transforming a simulation-sound protocol into a simulation-extractable one, relying only on stand-alone semi-honest oblivious transfer (sh-OT) protocols. Our compilation technique uses similar ideas as that in [21, 22] that achieves extractability using OT; furthermore, interestingly, though our compilation technique is non-black-box, it is inspired by the black-box compilation technique used in [37, 38] for transforming a sh-OT protocol into one secure against malicious adversaries (m-OT protocol). At a very high-level, we use the idea of having an OT execution with two random inputs at the prover's side (acting as the sender) and fixed input index 1 at the verifier's side (acting as the receiver), and later letting the prover use the second random input to hide the witness. This idea leads to a simple protocol as, even if the verifier deviates from the honest behavior in the OT execution, it learns no information of the witness; therefore, it suffices to require the verifier to prove of its honest behavior after the OT execution (instead of giving a proof after every message in the OT execution as the standard technique requires). Next we explain our compilation technique in more details.

First, it follows from standard techniques that the existence of a sh-OT protocols implies the existence of a full-fledged OT protocol against malicious adversaries (m-OT for short). Then given a simulation-sound ZK (ssZK) protocol, our

[11] The actually protocol is more complicated, as the notion of $\mathcal{SNMWI}$ arguments are only defined with respect to languages with unique witness. But for an intuitive explanation of high-level ideas here, we omit the complication.

compilation technique outputs a protocol $\langle P, V \rangle$ as follows: In the first stage, the prover and the receiver participates in an execution of a m-OT protocol where the prover acts as the OT sender using two random inputs r_1 and r_2 and the verifier acts as the OT receiver choosing the first input; in the second stage, the verifier proves that it has used input index 1 in the OT execution using the ssZK protocol; if the proof is accepting, the prover then sends the witness w padded with the second random input $w \oplus r_2$ in the third stage, followed by a proof in the fourth stage that this message XOR'ed with the second sender's input in the OT execution is indeed a valid witness of the statement being proved using again the ssZK protocol. The high level idea of the protocol $\langle P, V \rangle$ is simple. First of all, it is concurrently simulatable: To simulate a proof of statement x, a simulator can send a random string in the third stage in place of $w \oplus r_2$ and "cheats" in the proof in the last stage by relying on the concurrent-simulation property of the ssZK protocol; (it acts honestly in the first two stages). To see that $\langle P, V \rangle$ is further concurrently simulation-extractable, consider a man-in-the-middle adversary that receives many proofs, referred to as the left proofs, and gives many proofs, referred to as the right proofs, concurrently. We construct a simulator-extractor (which eventually corresponds to the simulator of our UC secure protocols) that concurrently simulates all the left proofs as described above and extracts a witness from every convincing right-proof as follows: In a right proof, the simulator-extractor (acting as the verifier) chooses the *second input* in the OT execution and "cheats" in the proof in the second stage relying again on the concurrent simulation property of the ssZK protocol; it then recovers the witness by simply XORing the third stage message with the second input it obtains in the OT execution. To show that simulator-extractor always extracts valid witnesses from the adversary, it boils down to show that the adversary is never able to prove a false statement using the ssZK protocol, even amid simulation, which essentially relies on the simulation-soundness property of the ssZK protocol.

However, some subtleties arise: The simulator-extractor simulates for the adversary both proofs of the ssZK protocol and OT executions. The simulation-soundness property only guarantees that the adversary cannot cheat when receiving simulated proofs of the ssZK protocols, but not simulated OT executions. (This problem is in the same spirit as the problems encountered in [39–41] when using non-malleable commitments as a sub-protocol in a larger protocol.) To solve this problem, we enhance the security of our ssZK protocol so that it is also simulation-sound w.r.t. the OT protocol—namely, even when the adversary receives many simulated executions of the OT protocol, it still cannot prove any false statement. In fact, we will design a protocol that is simulation-sound both w.r.t. itself and to any protocols with a fixed bounded number of rounds; this is achieved by relying on a notion of k-robust $\mathcal{SNMWI}$ protocol, which is a $\mathcal{SNMWI}$ protocol that additionally guarantees that no adversary can correlates the witness it uses in a proof with the "secret" in a k-round interaction it participates in, provided that messages in that interaction are indistinguishable (when generated with different secrets). This notion is in analogy to the notion

of k-robust non-malleable commitments [40]; and as we show, can be realized using a k-robust non-malleable commitment scheme. Then since as shown in [40], k-robust non-malleable commitment can be constructed from the minimal assumption of OWF, so can k-robust $\mathcal{SNMWI}$ protocols. Finally, we remark that this problem of robustness is not present in [1]; there, the compilation technique of [25, 36, 33] only implicitly requires the ssZK protocol to be simulation-sound w.r.t. non-interactive protocols, which is satisfied by any ssZK protocol that is an argument of knowledge (as required by the compilation technique).

An additional issue that we encounter is that for the above argument to go through, we need the OT protocol to satisfy some additional properties. More precisely, recall that the proof of concurrently simulatability of $\langle P, V\rangle$ requires showing that as long as the adversary can prove that it has acted honestly in the OT execution with input 1, the sender's second random input is completely hidden. At a first glance, it seems that this follows directly from the security against malicious receiver of the OT protocol. However, it may be possible for a malicious receiver to obtain the second input in the OT execution, but later explain its behavior with input 1. Fortunately, the security property that we need is exactly captured by the notion of *defensible privacy for the receiver* introduced by [37], which, roughly speaking, ensures that as long as a malicious receiver can output a good "defense"—that is, explaining its behavior as an honest receiver with input b and random tape σ—at the end of the OT execution, then the honest sender's other input $b' \neq b$ must remain hidden. Furthermore, to show that $\langle P, V\rangle$ is simulation extractable, we need the OT protocol to satisfy that as long as a malicious sender can output a good "defense", with inputs r_1, r_2 and random tape σ', after an OT execution, the honest receiver with input b must obtain r_b. To formalize this security property, we adapt the notion of defensible privacy of [37] to consider the correctness requirement; we called it the *defensible correctness property*. Therefore, our compilation technique relies on a m-OT protocol that is defensibly private for the receiver and defensibly correct for the sender; we show that such a protocol is implied by the existence of sh-OT protocols.

Constant-round SPS-security from polynomial-time hard sh-OT: In [1], the authors constructed SPS-secure protocols with strong indistinguishability: Real-world executions of these protocols are indistinguishable to ideal-world simulations, against distinguishers of the same time complexity of the simulator, which is super-polynomial. To obtain a model of security that can be implemented in constant rounds from standard polynomial time harness assumptions (in the plain model), we weaken the generalized model of UC security in [1] to require only *plain indistinguishability* against $\mathcal{PPT}$ distinguishers. However, even with this weakening, at the first glance, it is still unclear how to achieve plain-SPS-security from only polynomial time hardness assumptions. Let us illustrate the difficulty using the above described protocol $\langle P, V\rangle$ that implements the ideal ZK functionality.

In order to simulate the view of and extract witnesses from a man-in-the-middle adversary, the simulator-extractor of $\langle P, V\rangle$ simulates all the ssZK proofs

to the adversary, as well as all the OT executions it participates in. The latter can be simulated efficiently, but, the concurrent simulation of the ssZK arguments takes super-polynomial time in the SPS-model. Then it seems that in order to apply the security guarantees of the sh-OT protocol and the simulation soundness property of the ssZK protocol (to show that the view of the adversary is indistinguishable and it never proves any false statement), we need the security of the sh-OT and ssZK protocols to hold against super-polynomial machines, (since the adversary, though a $\mathcal{PPT}$ machine itself, receives many simulated proofs generated in super-polynomial time). Roughly, this is the technical reason why the LPV protocol relies on super-poly hardness assumption.

To get around this problem, we exploit the structure of the ssZK protocol constructed in [1]. Recall that it consists of a UC-puzzle execution where the verifier establishes a trapdoor, followed by a proof using the $\mathcal{SNMWI}$ argument that either the statement is true or a trapdoor is known. The key observation is that when simulating a proof of this protocol, only the simulation of the UC puzzle takes super-polynomial time; once a trapdoor is obtained, the rest of the simulation can be done efficiently. Therefore, if we modify the protocol $\langle P, V\rangle$ to have the puzzle executions in the two ssZK proofs sent at the beginning of the protocol—call it the preamble phase of the protocol—we obtain a protocol $\langle P', V'\rangle$ that has the same property: Only the preamble phase of the protocol takes super-polynomial time to simulate (the rest can be simulated efficiently given the puzzle answers). With this simple change, now we only need the sh-OT and $\mathcal{SNMWI}$ protocols to be secure for polynomial-time. To illustrate our idea, consider first the stand-alone setting. To show that $\langle P', V'\rangle$ is zero-knowledge, we rely on the "hiding" property of the sh-OT and the $\mathcal{SNMWI}$ protocols; since the simulation of the preamble phase happens before them, and thus the puzzle answers can be fixed non-uniformly, it suffices to rely on "hiding" against non-uniform $\mathcal{PPT}$ machines.

We use the same idea to prove the concurrent security of $\langle P', V'\rangle$: Establish the simulation-extractability property of $\langle P', V'\rangle$ in a sequence of hybrids that gradually simulate each session in two steps (the preamble phase first and then the rest) in a clever order. More precisely, consider a man-in-the-middle adversary that participates in m proofs; order all the proofs according to the sequence in which their preamble phases *completes*. Then consider a sequence of $2m + 1$ hybrids $H_0, \ldots, H_{2m+1}$'s, where in hybrid H_{2i} the first i sessions are simulated, and in hybrids H_{2i+1} and $H_{2(i+1)}$ (in addition to the first i sessions) the preamble phase and the rest of the $(i + 1)^{\text{th}}$ session are simulated respectively. To show that $\langle P', V'\rangle$ is simulation-extractable, it boils down to prove that every two subsequent hybrids are indistinguishable and the adversary never proves a false statement using the $\mathcal{SNMWI}$ argument in all hybrids. From hybrid H_{2i} to H_{2i+1} this follows directly from the statistical simulation property of the UC-puzzles. From hybrid H_{2i+1} to $H_{2(i+1)}$, this relies on the security of the sh-OT and $\mathcal{SNMWI}$ protocol executions in the $(i + 1)^{\text{th}}$ session; since in these two hybrids, only puzzles in the first $i + 1$ sessions are simulated, which happens before the OT and $\mathcal{SNMWI}$ executions in the $(i+1)^{\text{th}}$ session and can be fixed

non-uniformly, we only need the security of the OT and $\mathcal{SNMWI}$ protocols to hold against non-uniform $\mathcal{PPT}$ machines. Given that $\mathcal{SNMWI}$ arguments are implied by sh-OT protocols, $\langle P', V' \rangle$ implements the ideal ZK functionality with plain-SPS-security based on only polynomial-time hard sh-OT protocols.

Now, it seems that by simply combining $\langle P', V' \rangle$ with previous constructions of UC secure protocols Π that uses the ideal ZK functionality $\mathsf{IdealZK}$ [25, 33, 36, 9, 4], we can obtain constant-round plain-SPSsecure computation from sh-OT protocols. Unfortunately, previous constructions rely on the existence of sh-OT protocols; if composing them with $\langle P', V' \rangle$ in the straightforward way—replacing every $\mathsf{IdealZK}$ call in Π with an invocation of $\langle P', V' \rangle$—for the composed protocol $\Pi' = \Pi^{\mathsf{IdealZK}/\langle P', V' \rangle}$ to be secure in general, we need Π to be secure against super-poly time, which requires super-poly hard sh-OT! To resolve this problem, we modify the composed protocol Π' as we did to the protocol $\langle P, V \rangle$: Consider a protocol Π'' that is identical to Π' except that all the puzzle-executions in the invocations of $\langle P', V' \rangle$ are executed in parallel at the beginning of the protocol, call this again the preamble phase of the protocol; now Π'' has the property that only its preamble phase takes super-polynomial time to simulate, and the rest can be simulated efficiently with puzzle answers. Therefore, by considering a similar sequence of hybrids as in the proof of $\langle P', V' \rangle$, we can prove the security of Π'' directly.

3 UC Security Based on Stand-Alone Semi-Honest OT

We consider the $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-UC-model introduced in [1]. The model extends the framework of universal composability [5]. The key difference from UC lies in that in UC, the environment is modeled as a non-uniform $\mathcal{PPT}$ machine and the ideal-model adversary (or simulator) as a uniform $\mathcal{PPT}$ machines, whereas in the general model, the environment and the simulator are allowed to be from arbitrary complexity classes $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$ respectively. (Note, however, that the adversary is still uniform $\mathcal{PPT}$.) One important affect of this change is that the UC composition theorem [5] no longer holds; as a result, the stand-alone security of a protocol does not directly imply the concurrent security. In remedy, in the general model, an environment executing a protocol π can start many instances of the protocol, and thus implementing a functionality $\mathcal{F}$ in the general model means directly implementing the multi-session extension $\hat{\mathcal{F}}$[12] of $\mathcal{F}$. We focus only on static adversaries. Let $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$ represent the closure of $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$ that includes all computations by $\mathcal{PPT}$ oracle Turing machines M with oracle access to $\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}}$. In this section, we prove the following main technical theorem.

Theorem 3. *Assume the existence of a t_P-round $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure UC-puzzle in a $\mathcal{G}$-hybrid model, and a t_{OT}-round stand-alone sh-OT protocol secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$. Then, for every "well-formed" functionality $\mathcal{F}$, there exists a $O(t_P + t_{OT})$-round protocol Π in the $\mathcal{G}$-hybrid model that $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-UC-realizes $\mathcal{F}$.*

[12] Informally speaking, $\hat{F}$ emulates many independent copies of $\mathcal{F}$ running concurrently; see [9, 1] for a formal definition.

3.1 Proof of Theorem 3

Recall that the $\mathsf{IdealZK}$ functionality parameterized with a language L implements the function $\mathsf{ZK}^L((x,w),x) = (\perp, b)$, where $b = 1$ if w is a valid witness for the membership of x in L and 0 otherwise. Then Theorem 3 follows from the following two lemmas.

Lemma 1 ($\mathsf{IdealZK}$-Lemma) *Assume the existence of t-round stand-alone secure sh-OT secure w.r.t $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$. Then, for every well-formed functionality $\mathcal{F}$, there exists a $O(t)$-round protocol Π in the ZK-Hybrid model that $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-UC-realizes $\mathcal{F}$.*

Lemma 2 (Puzzle-Lemma) *Let Π' be a protocol in the $\mathsf{IdealZK}$ model. Assume the existence of a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure t_P-round puzzle $\langle S, R\rangle$ in a $\mathcal{G}$-hybrid model, a t_{OT}-round stand-alone sh-OT protocol $\langle S_{OT}, R_{OT}\rangle$ that is secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$, and a t_{WI}-round t_{OT}-robust $\mathcal{SNMWI}$ protocol $\langle P_s, V_s\rangle$ secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$. Then, there exists a $O(t_P + t_{WI} + t_{OT})$-round protocol Π in the $\mathcal{G}$-hybrid that $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-UC emulates Π'.*

The first lemma is implicit in previous works [25, 42, 4, 9] for normal UC-security (i.e., (n.u.$\mathcal{PPT}$, $\mathcal{PPT}$)-UC-security) and can be easily extended to the general $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-UC model assuming stand-alone sh-OT protocol secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$; we omit the proof (see [1] for a similar proof assuming TDP's). Next, towards proving the puzzle lemma, we provide a general transformation that transforms any protocol Π in the ZK-Hybrid model into a protocol Π' in the real model using a special-purpose zero-knowledge protocol that is "concurrently simulatable" and "concurrently simulation-extractable".

Special-purpose ZK Protocol $\langle P, V\rangle$. The construction of $\langle P, V\rangle$ relies on the following three building blocks; all with security against class $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$. (1) A t'-round m-OT protocol $\langle S_{OT}, R_{OT}\rangle$ that is defensibly private for the receiver and defensibly correct for the sender; it follows from standard techniques [4, 43] that such a protocol exists assuming t_{OT}-round sh-OT protocols, and $t' = O(t_{OT})$. (2) A t'-robust $\mathcal{SNMWI}$ protocol $\langle P_s, V_s\rangle$; it follows from a similar proof as in [1] that such a protocol exists assuming OWF's and the round-complexity is of $O(t')$. (We defer the formal construction and proof of such m-OT and robust $\mathcal{SNMWI}$ to the full version.) (3) A $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure puzzle $(\langle S, R\rangle, \mathcal{R})$ in a $\mathcal{G}$ hybrid model. For simplicity of exposition, our description below rely on a statistically binding commitment scheme com that has *unique decommitment*, that is the transcript of the commitment not only uniquely decides the value committed to inside but also the decommitment with overwhelming probability; but the protocol can be easily modified to work with any arbitrary statistically binding commitment (see the full version for more details). Then, the special-purpose ZK protocol $\langle P, V\rangle$ for a **NP** relation R_L proceeds as follows: To prove a statement x, the prover and verifier with identities id_P and id_V, and additional auxiliary input $w = R_L(x)$ for the prover, interacts in six stages.

Stage 1: The Prover and Verifier participate in a puzzle-interaction where the Verifier assumes the role of the sender and the Prover as the receiver. Let $\text{TRANS}_{V\to P}$ be the transcript of the messages exchanged.

Stage 2: The Prover and Verifier participate in a second puzzle-interaction with the roles reversed, i.e. the Prover is the sender and the Verifier is the receiver. Let $\text{TRANS}_{P\to V}$ be the transcript of the messages exchanged.

Stage 3: The Prover first selects two random string $r_1, r_2 \in \{0,1\}^n$. Then the Prover and Verifier interact using $\langle S_{OT}, R_{OT}\rangle$, where the Prover is the sender with inputs (r_1, r_2) and the Verifier is the receiver with input 1. Let TRANS_{OT} be the transcript of the messages exchanged.

Stage 4: The Verifier commits to s using com. Then it proves using the protocol $\langle P_s, V_s\rangle$ and identity id_V, the statement that it either committed to a string s that contains a valid witness establishing the verifiers input as index 1 in TRANS_{OT} and the string output by the receiver at the end of the Stage 3 protocol or a string s such that $(s, \text{TRANS}_{P\to V}) \in \mathcal{R}$.

Stage 5: The Prover sends the string $r = r_2 \oplus w$ in the clear.

Stage 6: The Prover commits to s' using com. Then the prover proves using the protocol $\langle P_s, V_s\rangle$ and identity id_P, the statement that it either committed to a string s' that establishes that the inputs used by the prover in TRANS_{OT} is (r_1, r_2) such that $r_2 \oplus r \in R_L(x)$ or a string s' such that $(s', \text{TRANS}_{V\to P}) \in \mathcal{R}$.

Realizing the IdealZK-functionality: Given any protocol Π' in ZK-Hybrid model and the special-purpose zero-knowledge protocol $\langle P, V\rangle$, the protocol Π in the real model is constructed from Π' by instantiating the IdealZK functionality using $\langle P, V\rangle$. All invocations of the IdealZK functionality in which P_i provers to P_j a statement x using witness w is replaced with an subroutine call of $\langle P, V\rangle$ between P_i and P_j where P_i proves the statement x using witness w to P_j, using identities $\mathsf{id}_P = i$ and $\mathsf{id}_V = j$ respectively. Due to the lack of space, we defer the formal security proof of Π that it emulates Π' in the ZK-Hybrid model to the full version.

References

1. Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: universal composability from stand-alone non-malleability. In: STOC. (2009) 179–188
2. Nguyen, M.H.: The relationship between password-authenticated key exchange and other cryptographic primitives. In: TCC. (2005) 457–475
3. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: FOCS. (1986) 162–167
4. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC. (1987) 218–229
5. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS. (2001) 136–145
6. Canetti, R., Fischlin, M.: Universally composable commitments. In: CRYPTO. (2001) 19–40

7. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: EUROCRYPT. (2003) 68–86
8. Lindell, Y.: General composition and universal composability in secure multi-party computation. In: FOCS. (2003) 394–403
9. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: STOC. (2002) 494–503
10. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: FOCS. (2004) 186–195
11. Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: TCC. (2007) 61–85
12. Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent general composition of secure protocols in the timing model. In: STOC. (2005) 644–653
13. Canetti, R., Pass, R., Shelat, A.: Cryptography from sunspots: How to use an imperfect reference string. In: FOCS. (2007) 249–259
14. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: EUROCRYPT. (2003) 160–176
15. Prabhakaran, M., Sahai, A.: New notions of security: achieving universal composability without trusted setup. In: STOC. (2004) 242–251
16. Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In: FOCS. (2005) 543–552
17. Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. In: FOCS. (2006) 367–378
18. Goldreich, O.: Foundations of Cryptography — Basic Applications. Cambridge University Press (2004)
19. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC. (1988) 20–31
20. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: CRYPTO. (2008) 572–591
21. Damgård, I., Nielsen, J.B., Orlandi, C.: On the necessary and sufficient assumptions for uc computation. In: TCC. (2010) 109–127
22. Maji, H.K., Prabhakaran, M., Rosulek, M.: A zero-one law for cryptographic complexity with respect to computational uc security. In: CRYPTO. (2010) 595–612
23. Goyal, V.: Constant round non-malleable protocols using one way functions. In: STOC. (2011) 695–704
24. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. J. ACM **51**(6) (2004) 851–898
25. Pass, R.: Bounded-concurrent secure multi-party computation with a dishonest majority. In: STOC, New York, NY, USA, ACM (2004) 232–241
26. Katz, J.: Which languages have 4-round zero-knowledge proofs? In: Theory of Cryptography. (2008) 73–88
27. Groth, J., Ostrovsky, R.: Cryptography in the multi-string model. In: CRYPTO. (2007) 323–341
28. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: TCC. (2010) 308–326
29. Damgård, I., Groth, J.: Non-interactive and reusable non-malleable commitment schemes. In: STOC. (2003) 426–437
30. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: FOCS. (2010) 541–550
31. Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: EUROCRYPT. (2012) 99–116

32. Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS. Volume 0. (2001) 106–115
33. Pass, R., Rosen, A.: Bounded-concurrent secure two-party computation in a constant number of rounds. In: FOCS. (2003) 404–
34. Lindell, Y.: Lower bounds and impossibility results for concurrent self composition. J. Cryptology **21**(2) (2008) 200–249
35. Goyal, V., Jain, A., Ostrovsky, R.: Password-authenticated session-key generation on the internet in the plain model. In: CRYPTO. (2010) 277–294
36. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: STOC. (2003) 683–692
37. Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions for secure computation. In: STOC. (2006) 99–108
38. Haitner, I.: Semi-honest to malicious oblivious transfer - the black-box way. In: TCC. (2008) 412–426
39. Lin, H., Pass, R., Tseng, W.L.D., Venkitasubramaniam, M.: Concurrent non-malleable zero knowledge proofs. In: CRYPTO. (2010) 429–446
40. Lin, H., Pass, R.: Non-malleability amplification. In: STOC. (2009) 189–198
41. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. To appear in FOCS 2010 (2010)
42. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: STOC. (1990) 503–513
43. Goldreich, O.: Foundations of Cryptography — Basic Tools. Cambridge University Press (2001)