Interactive Locking, Zero-Knowledge PCPs, and Unconditional Cryptography*

Vipul Goyal^{1,**}, Yuval Ishai^{2,***}, Mohammad Mahmoody^{3,†}, and Amit Sahai^{4,‡}

[1] Microsoft Research, India
vipul@microsoft.com
[2] Technion and UCLA
yuvali@cs.technion.ac.il
[3] Princeton University
mohammad@cs.princeton.edu
[4] UCLA
sahai@cs.ucla.edu

Abstract. Motivated by the question of basing cryptographic protocols on stateless tamper-proof hardware tokens, we revisit the question of unconditional two-prover zero-knowledge proofs for **NP**. We show that such protocols exist in the *interactive PCP* model of Kalai and Raz (ICALP '08), where one of the provers is replaced by a PCP oracle. This strengthens the feasibility result of Ben-Or, Goldwasser, Kilian, and Wigderson (STOC '88) which requires two stateful provers. In contrast to previous zero-knowledge PCPs of Kilian, Petrank, and Tardos (STOC '97), in our protocol both the prover and the PCP oracle are efficient given an **NP** witness.

Our main technical tool is a new primitive that we call *interactive locking*, an efficient realization of an unconditionally secure commitment scheme in the interactive PCP model. We implement interactive locking by adapting previous constructions of *interactive hashing* protocols to our setting, and also provide a direct construction which uses a minimal amount of interaction and improves over our interactive hashing based constructions.

Finally, we apply the above results towards showing the feasibility of basing unconditional cryptography on *stateless* tamper-proof hardware tokens, and obtain the following results. **(1)** We show that if tokens can be used to encapsulate other tokens, then there exist unconditional and statistically secure (in fact, UC secure) protocols for general secure

* The full version of the paper is available at: http://eprint.iacr.org/2010/089

** This work was done mostly while this author was at UCLA, supported in part from NSF grants listed below.

*** Supported in part by ISF grant 1310/06, BSF grants 2004361, 2008411, and NSF grants 0716835, 0716389, 0830803, 0916574.

† This work done partially while this author was visiting UCLA. Supported by NSF grants 0627526, 0426582 and 0832797.

‡ Supported in part by BSF grants 2004361, 2008411, and NSF grants 0916574, 0716389, 0627781, 0830803.

T. Rabin (Ed.): CRYPTO 2010, LNCS 6223, pp. 173–190, 2010.

computation. **(2)** Even if token encapsulation is not possible, there are unconditional and statistically secure commitment protocols and zero-knowledge proofs for **NP**. **(3)** Finally, if token encapsulation is not possible, then no protocol can realize statistically secure oblivious transfer.

1 Introduction

What is the minimal amount of trust required for unconditionally secure cryptography? Unconditional cryptography can be based on trusted two-party functionalities such as oblivious transfer [1,2] or noisy channels [3], on bounded storage assumptions [4], on the presence of an honest majority [5,6,7], or even on the presence of a dishonest majority of *non-communicating* parties [8]. More recently, there has been a considerable amount of work on cryptographic protocols in which parties can generate and exchange tamper-proof hardware tokens. In this model it was shown that unconditionally secure commitments [9] or even general secure two-party computation [10] are possible, provided that the tokens can be *stateful*. In particular, stateful tokens can erase their secrets after being invoked. The present work is motivated by the goal of establishing unconditional feasibility results for cryptography using *stateless* hardware tokens. This question turns out to be related to the classical question of unconditional multi-prover zero-knowledge proofs, which we revisit in this work. We start with some relevant background.

Multi-Prover Zero-Knowledge. Since the introduction of zero-knowledge proofs in the seminal work of Goldwasser, Micali, and Rackoff [11], a large body of work has been devoted to understanding the capabilities and limitations of such proofs. A particularly successful line of research studied the power of *statistical zero-knowledge* (SZK) proofs — ones which guarantee that even computationally unbounded verifiers can learn nothing from the interaction with the prover. In contrast to computational zero-knowledge proofs [12], a major limitation of SZK proofs which restricts their usefulness in cryptography is that they seem unlikely to cover the entire class of **NP** [13,14]. The related goal of obtaining any kind of *unconditional* zero-knowledge proofs for **NP**, which do not rely on unproven intractability assumptions, seems as unlikely to be achieved (cf. [15]) at least until the elusive **P** vs. **NP** question is resolved.

Motivated by the above goals, Ben-Or, Goldwasser, Kilian, and Wigderson [8] introduced in 1988 the model of *multi-prover interactive proofs* (MIPs), a natural extension of the standard model of interactive proofs which allows the verifier to interact with two or more non-communicating provers. The main result of [8] is an unconditional two-prover SZK proof for any language in **NP** (see [16,17,18] for subsequent improvements). A direct cryptographic application suggested in [8] is that of proving one's identity using a pair of bank cards. We will further discuss these types of applications later.

In a very surprising turn of events, the initial work on zero-knowledge in the MIP model led to a rapid sequence of developments that have literally transformed the theory of computer science. This line of research culminated in the first proof of the PCP Theorem [19,20].

The notion of probabilistically checkable proofs (PCPs) is very relevant to our work. In 1988, Fortnow, Rompel, and Sipser [21] suggested an alternative model for MIPs in which multiple provers are replaced by a single oracle, subsequently called a *PCP oracle* or just a PCP. The difference between an oracle and a prover is that an oracle, like a classical proof, cannot keep an internal state. When a prover is asked multiple queries, the answer to each query can depend on all previous queries, whereas the answer of an oracle to each query must depend on that query alone. The latter difference makes soundness against PCP oracles easier to achieve than soundness against provers, which explains the extra power of PCPs over traditional interactive proofs. However, as already observed in [8], the *zero-knowledge* property becomes harder to achieve when converting provers into oracles because oracles have no control over the number of queries made by a dishonest verifier. In particular, if the verifier may query the entire domain of the oracle (as in the case of traditional polynomial-length PCPs) then the oracle can no longer hide any secrets.

The question of replacing zero-knowledge provers by stateless oracles is motivated by practical scenarios in which verifiers can "reset" provers to their initial state, say by cutting off their power supply. (Note that similarly to zero-knowledge provers, zero-knowledge PCP oracles should be *randomized* in the sense that their answer depends both on the query and on a secret source of randomness which is picked once and for all when the oracle is initialized.) This motivation led to a recent line of work on *resettable zero-knowledge*, initiated by Canetti, Goldreich, Goldwasser, and Micali [22]. The main results from [22] show that, under standard cryptographic assumptions, there exist resettable (computational) zero-knowledge proofs for **NP**. However, results along this line do not seem relevant to the case of *unconditional* (and statistical) zero-knowledge proofs, which are the focus of the present work.

Zero-knowledge PCPs. The question of *unconditional* zero-knowledge PCPs was studied by Kilian, Petrank and Tardos [23] (improving over previous results implicit in [18]). Specifically, it is shown in [23] that any language in **NEXP** admits a proof system with a *single* PCP which is statistical zero-knowledge against verifiers that can make any polynomial number of PCP queries (but are otherwise computationally unbounded). However, as expected from proof systems for **NEXP**, the answers of the PCP oracle cannot be computed in polynomial time. This still leaves hope for scaling down the result to **NP** and making the PCP oracle efficient given an **NP** witness. Unfortunately, such a scaled down version presented in [23] has the undesirable side effect of scaling down the zero-knowledge property as well, effectively restricting the number of queries made by a cheating verifier to be much smaller than the (fixed polynomial) entropy of the oracle. Thus, compared to typical feasibility results in cryptography, the results of [23] for **NP** require us to either make an unreasonable assumption about the computational capability of the (stateless) prover, or to make an unreasonable assumption about the limitations of a cheating verifier.

Interactive PCPs. The above state of affairs motivates us to consider the *Interactive PCP* (IPCP) model, which was recently put forward by Kalai and Raz [24] and further studied in [25]. This model can be seen as lying in between the pure PCP model and the pure MIP model, thus aiding us in our quest for a "minimal" model for efficient unconditional zero-knowledge proofs for **NP**. In the IPCP model there is one interactive prover as in the MIP model and one PCP as in the PCP model. The study of IPCPs in [24] was motivated by the efficiency goal of allowing *shorter* PCPs for certain **NP** languages than in the traditional PCP model, at the price of a small amount of interaction with a prover. In contrast, our use of the IPCP model is motivated by the *feasibility* goal of obtaining unconditional zero-knowledge proofs for **NP** with polynomial-time prover and PCP oracle. Another difference is that while in the context of [24] a PCP is at least as helpful as a prover, the zero-knowledge property we consider is harder to satisfy with a PCP oracle than with a prover (as discussed above). The IPCP model can be made strictly stronger than the MIP model by requiring soundness to hold also with respect to *stateful* PCP oracles. We tackle this stronger variant as well, but we stick to the basic IPCP model by default.

To meaningfully capture zero-knowledge proofs with polynomial-time provers in the IPCP model, we extend the original IPCP model from [24] in two natural ways. First, we allow the PCP to be randomized. Concretely, we assume that both the prover and the PCP are implemented by polynomial-time algorithms with three common inputs: an instance x, a witness w, and a random input r. (This is analogous to earlier models for efficient multi-prover zero-knowledge proofs for **NP**.) The length of both w and r is polynomial in $|x|$. Second, as discussed above, in order to allow the PCP oracle to hide secrets from the verifier we need to use PCP oracles with a super-polynomial query domain, and we restrict cheating verifiers to make (an arbitrary) polynomial number of queries to the oracle, but otherwise allow them to be computationally unbounded. Note, however, that in contrast to the solutions from [23] we cannot use PCP oracles with a super-polynomial entropy since we want our PCP to be efficiently implementable.

This gives rise to the following feasibility question:

> *Are there (efficient-prover) statistical zero-knowledge proofs for* **NP** *in the interactive PCP model?*

Our Results. We answer the above question affirmatively, presenting an *unconditional* SZK proof for **NP** in the interactive PCP model with efficient prover and PCP oracle. Zero-knowledge holds against cheating verifiers which can make any polynomial (in fact, even sub-exponential) number of PCP queries, but are otherwise computationally unbounded. Our protocol can be implemented in a constant number of rounds. We also show how to get a similar protocol (with a non-constant number of rounds) in the stronger variant of the IPCP model in which a cheating PCP oracle may be stateful, thus strengthening the previous feasibility result from [8].

Interactive locking. The main technical tool we use to obtain the above results (as well as additional applications discussed below) is a new primitive which we call an *interactive locking scheme* (ILS). This primitive extends in a natural way the notion of non-interactive locking schemes which were defined and implemented in [23]. The original locking primitive can be viewed as a PCP-based implementation of a non-interactive commitment with statistical hiding and binding. Roughly speaking, a locking scheme is an oracle which hides a secret that can later be revealed to the receiver by sending it a decommitment key. Given access to the oracle alone, it is hard for the receiver to learn anything about the secret. However, it is easy for the receiver to become convinced that at most one secret can be successfully decommitted even when the oracle is badly formed.

The locking scheme from [23] requires the oracle to have bigger entropy than the number of queries against which the hiding property should hold. We prove the intuitive fact that such a limitation is inherent, and therefore there is no efficient-oracle non-interactive locking scheme which resists an arbitrary polynomial number of queries. This is because intuitively if the entropy of the oracle is bounded, then either: (1) the receiver is able to learn all the entropy by making a polynomial number of queries, and therefore break the hiding property; or (2) if some entropy is hidden no matter what queries the receiver makes, then a cheating sender is able to create a "fake" oracle that can cheat on this entropy and therefore be opened to any value, breaking the binding property.

This motivates our notion of an *interactive* locking scheme. An ILS is a locking scheme in the IPCP model: the commitment phase can involve, in addition to oracle queries by the receiver, interaction with the sender from whom the secret originated. Here the sender and the oracle play the roles of the prover and PCP oracle in the IPCP model, respectively. Decommitment still involves a single message from the sender to the receiver. Somewhat surprisingly (and counter to our own initial intuition), we show that interaction can be used to disrupt the intuitive argument above.

We present several constructions of efficient interactive locking schemes. We show how to obtain such schemes from *interactive hashing* — a primitive which was introduced by Naor, Ostrovsky, Venkatesan, and Yung [26] for the purpose of constructing statistically hiding and *computationally* binding commitment schemes from any one-way permutation (see also [27,28,29]). The high level idea of the transformation from interactive hashing to ILS is to "implement" a one-way permutation by an oracle which contains a random *point function* (i.e., a function that outputs 0 on all but one random point). To ensure the binding property even when the oracle is badly formed, the receiver should query the oracle on a small number random points to verify that it is not "too far" from a point function. The (black-box) proof of security of the interactive hashing protocol implies (unconditional) proof of security for the ILS.

The above connection allows us to use interactive hashing protocols from the literature for obtaining interactive locking schemes, but leaves open the question of minimizing the amount of interaction with the sender. We resolve this question by presenting a novel direct construction of ILS which requires only a single round of interaction with the sender.

The high level idea behind our single round ILS is as follows. The oracle π constructed by the sender will be the zero function over $\{0,1\}^n$ except for an "interval" of size 2^{cn}. That is, $\pi(x) = 1$ for $a \leq x \leq a + 2^{cn}$ and $\pi(x) = 0$ elsewhere. Depending on whether the sender commits to zero or one, the interval will be planted in the first or second half of the oracle π. The position a of the interval will be revealed to the receiver in the decommitment phase. When $c < 1$, the interval size 2^{cn} will be small enough to prevent the receiver from finding the committed bit during the commitment phase. But now the sender is able to cheat by planting intervals in both the first and second half of π. To guarantee binding, we let the receiver ask a "challenge" question about the interval in such a way that the sender cannot find a *pair* of planted intervals in the first and second half of π with the same challenge answer. A natural idea is to use a pairwise independent function $h\colon \{0,1\}^n \to \{0,1\}^{dn}$ and ask the sender to reveal $h(a)$. The sender is able to plant at most $2^{(1-c)n}$ *separate* intervals in each half of π. Each of the intervals in the first and second half of π will have the same hashes with probability 2^{-dn}. Therefore if $2(1 - c) < d$, then with high probability over the choice of h the sender is *not* able to find two intervals with the same hash value $h(a)$ and thus gets committed to a fixed bit. But now the information revealed by $h(a)$ might help the receiver find a non-zero point in π and break the hiding property. We show how to modify the a known construction of pairwise independent hash functions to get another function which is still almost pairwise independent but has the additional property that the preimages of any hash value are "scattered" in the domain of the hash function. The latter property prevents the receiver from taking advantage of the knowledge of $h(a)$ to find where the interval is planted. Using this approach we simultaneously guarantee binding and hiding.

Cryptography using hardware tokens. The above study of zero-knowledge interactive PCPs and interactive locking schemes is motivated by a recent line of research on the capabilities of cryptographic protocols in which parties can generate tamper-proof hardware tokens and send them to each other. Katz [30] shows that, under computational assumptions, general *universally composable* (UC) secure two-party computation [31] is possible in this model if the tokens are allowed to be *stateful*, and in particular can erase their secrets after being invoked. It was subsequently shown that even *unconditional* security is possible in this model, first for the case of commitment [9] and then for general tasks [10]. See [32,33,34] and references therein for other applications of stateful tokens in cryptography.

Obtaining similar results using *stateless* tokens turns out to be more challenging. Part of the difficulty stems from the fact that there is no guarantee on the functionality of tokens generated by malicious parties — they may compute arbitrary functions of their inputs and may even carry state information from one invocation to another. It was recently shown in [10], improving on [35], that any *one-way function* can be used for basing (computationally) UC-secure two-party computation on stateless tokens. More practical protocols which satisfy weaker notions of security were given in [36]. These works leave open the question of obtaining a similar result *unconditionally*, and with *statistical* security. (To get around impossibility results in the plain model, the number of queries to a token should be polynomially bounded, but otherwise malicious parties may be computationally unbounded.) In fact, the constructions from [35,10,36] may lead to a natural conjecture that achieving statistical security in this setting is impossible, since in these constructions all the "useful information" contained in tokens can be learned by a computationally unbounded adversary using a polynomial number of queries.

However, similar to the case of ILS discussed above, the combination of stateless tokens and interaction turns out to be surprisingly powerful. As already alluded to in [8], MIP protocols can naturally give rise to protocols in the hardware token model. In our case, we implement the ILS (or IPCP) by having a single sender (prover) create a stateless tamper-proof hardware token which implements the PCP oracle and send it to the receiver (verifier). Applying this to our results, this directly gives rise to the first unconditionally secure commitment protocols and SZK proofs for **NP** using stateless tokens.

We show how this can be extended to general unconditionally secure (in fact, UC-secure) two-party computation if parties are allowed to build tokens which encapsulate other tokens: namely, the receiver of a token A is allowed to build another token B which internally invokes A. The high level idea is the following. By the completeness of oblivious transfer (OT) [2,37], it suffices to realize OT using stateless tokens. This is done as follows. The OT sender's input is a pair of strings (s_0, s_1) and the OT receiver's input is a selection bit b. The OT receiver commits b using an ILS. Applying our best construction, this involves sending a token A to the OT sender and responding to a random challenge message received from the OT sender. The OT sender now prepares and sends to the receiver a token B with the following functionality. Token B accepts a selection bit b along with a corresponding decommitment message. It checks that the decommitment is valid (this involves invocations of the token A, which token B encapsulates) and then returns the string s_b if decommitment was successful. The binding property of the ILS guarantees that the OT receiver can learn at most one string s_b. The hiding property of the ILS guarantees that the sender cannot learn b.

Interestingly, we also show a matching negative result: if token encapsulation is not allowed, then statistically secure OT is impossible. This holds even if both parties are guaranteed to follow the protocol except for making additional queries to tokens in order to learn information about the other party's input. The proof of this negative result employs a variant of the recent notion of accessible entropy from [38] and has the following high level intuition: In the standard model *without* tokens, one way to explain why statistical OT is not possible is to consider the randomness r_R of the receiver conditioned on the transcript τ of the protocol. If this conditional distribution reveals information about the receiver's choice b, then an unbounded sender can cheat by sampling from this distribution. But if not, then an unbounded receiver can cheat by sampling from this distribution for both values of b, and using the result to obtain both strings s_0 and s_1 of the sender.

In the token model, however, this situation is not symmetric, since the sender might not know what queries the receiver has asked from the tokens it holds (or vice versa). Informally, we define a protocol (A, B) to have *accessible entropy* if the parties can nevertheless (information theoretically) sample their randomness conditioned on the *other* party's view. If an OT protocol did have accessible entropy, then essentially the above impossibility argument would apply. (In contrast, the original definition of accessible entropy of [38] required that the parties could *efficiently* sample, since the focus in that work was on analyzing protocols secure against computationally bounded parties.)

The technical core of our impossibility result is the following technical lemma: For any protocol (A, B) in the stateless token model, there is another protocol (A', B') that differs from (A, B) only in that the parties ask (a polynomial number) more queries to the tokens that they hold. Furthermore, almost all the entropy of the new protocol (A', B') is accessible. This lemma allows us to carry out the intuition above and rule out statistically secure OT in the stateless token model.

Organization. In Section 2, we define the notions of zero-knowledge IPCPs and ILS, and show how to use ILS to build unconditional zero-knowledge IPCPs for **NP**. We also show that interaction is required for efficient ILS. In Section 3, we show how to construct ILS. In Section 4, we show the implications of our work on (unconditionally secure) cryptography with tamper-proof hardware tokens.

2 Statistically Zero-Knowledge IPCP for NP

Interactive PCPs (Definition 1 below) were first introduced in [24] and combine the notion of oracle algorithms with interactive algorithms. Here we define IPCPs in a general way, not only for the purpose of a proof system, but rather as a model of interaction consisting of two interactive algorithms and a prover. (This way we can define our notion of interactive locking schemes as a protocol in the IPCP model implementing the commitment functionality.)

Definition 1. *(Adapted from [24]) An* interactive probabilistically checkable proof *(IPCP) $\Gamma = (P, \pi, V)$ consists of an interactive algorithm P (the* prover*), an oracle π (the* PCP oracle*), and an interactive algorithm V (the* verifier*) such that:*

- *P and π share common randomness r_P, and V is given the randomness r_V.*
- *P, π, and V will be given an input x of length $|x| = n$. P and π may also receive a common private input w.*[1]
- *The PCP oracle π is a function of (r_P, x, w, q) where q is a query of the verifier V. Since (r_P, x, w) is fixed at the beginning of the protocol, we might simply use $\pi(q)$ to denote the answer to the query q.*

[1] For example when (P, π) are efficient and $L \in \mathbf{NP}$, w could be a witness for $x \in L$.

- *P and V^{π} engage in an interactive protocol during which V can query the PCP oracle π and at the end V accepts or rejects.*

By an efficient *IPCP we mean one in which the prover P, the PCP oracle π, and the verifier V run in polynomial time over the input length $|x| = n$.*

By the *round complexity* of an IPCP we mean the number of rounds of interaction between the verifier and the *prover* (and not the PCP oracle) where each round consists of a message from the verifier followed by a message from the prover. (See the full version of the paper for more discussion on this definition and a comprehensive elaboration on the IPCP model.)

Now we define the notion of a proof system in the IPCP model which directly incorporates the statistical zero-knowledge feature. We use a quantitative definition allowing us to speak about exponential zero-knowledge (rather than just super-polynomial security).

Definition 2 (SZK-IPCP for languages). *We say that $\Gamma = (P, \pi, V)$ is an SZK-IPCP for the language L with SZK $(u(n), \epsilon(n))$ and soundness $1 - \delta(n)$ if the following holds:*

- **Completeness:** *If $x \in L$, then* $\Pr[\langle P, V^{\pi}\rangle(x) = 1] = 1$.
- **Soundness:** *Γ has soundness $1 - \delta$ if for all $x \notin L$ and for any arbitrary prover $\widehat{P}$ and oracles $\widehat{\pi}$ it holds that* $\Pr[\langle \widehat{P}, V^{\widehat{\pi}}\rangle(x) = 1] \le \delta(n)$.
- **Statistical zero-knowledge (SZK):** *We say that the IPCP Γ is (u, ϵ)-SZK for L with a straight-line*[2] *simulator if there is a simulator* Sim *as follows. The (straight-line) simulator* Sim *interacts with a (potentially malicious) verifier $\widehat{V}$, while the simulator* Sim *receives* all *the queries of the the verifier (including both the queries asked from the prover and from the oracle) and responds to them. Since an unbounded verifier can ask arbitrary number of queries from its oracle, here we put a bound u on the number of oracle queries asked by $\widehat{V}$ and demand the following to hold: For any $v \le u$, if $\widehat{V}$ asks at most v oracle queries, then* Sim *runs in time* $\mathrm{poly}(n, v)$ *and produces a view for $\widehat{V}$ which is ϵ-close to the view of $\widehat{V}$ when interacting with (P, π).*

We simply call Γ an SZK-IPCP for L with security u, if Γ is $(1 - 1/u)$-(adaptively)-sound and $(u, 1/u)$-SZK.

Note that when $u(n)$ is super-polynomial, Definition 2 implies zero-knowledge against polynomial-time verifiers.

We prove that $2^{\Omega(n)}$-secure constant-round SZK-IPCPs exist for any language $L \in \mathbf{NP}$ where both the prover and the PCP oracle in our construction can be implemented efficiently given a witness w for $x \in L$.

Theorem 3 (Constant-round SZK-IPCP for NP). *For any language $L \in$ **NP** there exists a 2-round efficient SZK-IPCP Γ_{2R} for L with security $2^{\Omega(n)}$.*

[2] Since all of our simulators in this paper are straight-line, for sake of simplicity here we only describe how to define SZK for IPCPs with straight-line simulators.

The simulator of Γ_{2R} in Theorem 3 is straight-line and therefore by a result of [39], for a small enough constant c, a 2^{cn}-fold concurrent composition of Γ_{2R} remains $(2^{\Omega(n)}, 2^{-\Omega(n)})$-SZK if the inputs to the instances of Γ_{2R} are fixed in the beginning.

Ideas of the proof of Theorem 3. Our main step to prove Theorems 3 is to construct an "interactive locking scheme" (ILS) (Definition 5), a primitive corresponding to commitment schemes in the IPCP model. In Theorem 6 we present an ILS with optimal round complexity (i.e. one round). Then we feed our ILS (as a commitment scheme) into the well-known construction of [12] to achieve zero-knowledge for **NP** with non-negligible soundness. A classical way to amplify the soundness of proof systems (while keeping the round-complexity) in the standard model of interaction is to use parallel composition. Firstly we define parallel composition of IPCPs (see the full version) in a careful way and prove an optimal bound on how the soundness amplifies in such a parallel composition. The latter result is interesting on its own since the IPCP model lies in between the single-prover and the multi-prover models and it is known [21] that the parallel repetition does *not* amplify the soundness in a simple exponential form (as one would wish). Secondly, we show that although the parallel composition might hurt the zero-knowledge in general, by crucially using a special feature of our ILS called "equivocability" (see Definition 5) one can prove that SZK is preserved under parallel composition. Roughly speaking, an ILS is equivocable, if a malicious sender can efficiently decommit to any desired value by changing the content of the oracle *after* the commitment phase. See the full version for the full proof of Theorem.

We also show how to achieve a $2^{\Omega(n)}$-secure SZK-IPCP for any $L \in$ **NP** where the security holds even against *stateful* oracles. A stateful oracle can save a state and behave as maliciously as an interactive algorithm. Namely, the answers returned by a (malicious) stateful oracle can depend on the previous queries asked to the oracle as well as the other queries asked in the same "round" of queries. We call such IPCPs (secure against stateful oracles) *adaptively*-sound.

Theorem 4 (Adaptively-secure SZK-IPCP for NP). *There exists a (*poly(n)*-round) efficient SZK-IPCP Γ_{adap} for any $L \in$* **NP** *with* adaptive-*security $2^{\Omega(n)}$.*

Ideas of the proof of Theorem 4. To prove Theorem 4, we use ideas from [40] about converting multi-prover proof systems into an equivalent two-prover one (with non-negligible soundness) where the second prover is asked only one query. When a prover is asked only one query, it can be considered as an oracle. In our transformation to achieve adaptive security in the IPCP model, we use a similar compiler to that of [40] over the IPCP Γ_{2R} of Theorem 3 and crucially use the fact that Γ_{2R} is "public-coin" (i.e. the soundness holds even if the prover gets to see which oracle queries are asked). A public-coin IPCP is one which is sound even if the prover gets to see the oracle queries asked by the verifier. Finally we use sequential composition to amplify the soundness. See the full version of the paper for the full proof of Theorem.

3 Interactive Locking Schemes

An *Interactive locking scheme* is a commitment scheme implemented in the IPCP model. A similar definition appeared in [23] without the interaction (i.e. only with an oracle), but as we will see in Theorem 6 *non-interactive* locking schemes are inherently inefficient and therefore not as applicable in cryptographic settings.

Definition 5 (Interactive locking scheme). *Let $\Lambda = (S, \sigma, R)$ be an* efficient *IPCP (where we call S the sender, σ the locking oracle and R the receiver). Λ is called an* interactive locking scheme *(ILS) for the message space W_n if it of the following form:*

The common input is 1^n where n is the security parameter. (S, σ) receive a private input $w \in W_n$ which is called the committed message as well as the private randomness r_S. The receiver R gets the randomness r_R. The receiver R gets oracle access to the locking oracle σ and R^σ interacts with S in two phases: **(1)** *commitment phase and* **(2)** *decommitment phase. The decommitment phase consists of only one message from the sender S to the receiver R which includes the committed message w and the private randomness r_S used by S. Following this message the receiver R (perhaps after asking more queries from the oracle σ) accepts or rejects. We demand the following properties to hold:*

- **Completeness:** *For any $w \in W_n$ if all parties are honest the receiver always accepts.*
- **Binding:** *We define Λ to be $(1-\delta)$-binding if for any sender $\widehat{S}$ and any oracle $\widehat{\sigma}$, with probability at least $1-\delta$ over the interaction of the commitment phase there is at most one possible w such that $\widehat{S}$ can decommit to successfully.*
- **Hiding:** *Let $\widehat{R}$ be any malicious receiver who asks at most u oracle queries from σ, and let τ_w be the random variable which consists of the transcript of the interaction of R with (S, σ) till the end of the commitment phase when the committed message is $w \in W$. Λ is (u, ϵ)-hiding if for every such malicious receiver $\widehat{R}$ and every $\{w_1, w_2\} \subseteq W$ it holds that $\mathsf{SD}(\tau_{w_1}, \tau_{w_2}) \leq \epsilon$.*
- **Equivocability:** *Λ is equivocable if there is an efficient sampling algorithm* Sam *that given (τ, w) where τ is the transcript (including the oracle queries) of the commitment phase of $\langle S, \widehat{R}^\sigma \rangle$ (for an arbitrary receiver $\widehat{R}$) and any $w \in W$, $\mathsf{Sam}(\tau, w)$ outputs r according to the distribution $(r_S \mid \tau, w)$. Namely r is sampled according to the distribution of the private randomness r_S of (S, σ) conditioned on w being the committed message and τ being the transcript of the commitment phase.*

We simply call the ILS Λ u-secure if it is $(1 - 1/u)$-binding and $(u, 1/u)$-hiding. If $W = \{0, 1\}$, we call Λ a bit-*ILS.*

The following theorem presents an ILS with optimal round complexity.

Theorem 6. *(A round-optimal ILS) Let $\ell(n) = \mathrm{poly}(n)$, then*

1. *There exist an efficient ILS $\Lambda_{\mathsf{1R}} = (S, \sigma, R)$ for the message space $\{0, 1\}^\ell$ with security $2^{\Omega(n)}$ which has a commitment phase of only one round.*

2. *Any ILS with a noninteractive commitment phase needs an inefficient oracle σ and thus Λ has optimal round-complexity (as an efficient ILS).*

In the full version of the paper we give a general construction of ILS from any interactive hashing scheme with some minimal properties. Unfortunately nontrivial interactive hashing needs at least two rounds of interaction and thus this approach is incapable of giving us a round-optimal ILS. Due to space limit we refer the reader for this connection to the full version and here will only present the optimal construction.

Before proving Theorem 6 we need the following lemma whose proof is immediate.

Lemma 7. *For $n > m$ let $\mathcal{A}$ be the family of $n \times m$ Boolean matrices as follows. To get a uniform member of A, choose the first $n-m$ rows all at random, and take the last m rows to be an independently chosen at random conditioned on having* full rank *m. Then for any $0 \neq x \in \{0,1\}^n$, it holds that $\Pr_{A \leftarrow \mathcal{A}}[xA = 0] \leq 2^{-m}$ (and equivalently for any $x_1 \neq x_2 \in \{0,1\}^n$ and $y \in \{0,1\}^m$, it holds that $\Pr_{A \leftarrow \mathcal{A}}[x_1 A = x_2 A] \leq 2^{-m}$).*

Construction 8 (A 1-round ILS) *Suppose $b \in \{0,1\}$ is the private message given to sender and the oracle (S, σ), and suppose R is the receiver. Let $m = 3n/4$. Below we associate $\{0,1\}^n$ with the integers $[0, 2^n)$ and all additions and subtractions below are modulo 2^n.*

The commitment phase of Λ_{1R}:

1. *Sender S chooses $a \leftarrow \{0,1\}^n$ at random. Let f_b be the function: $f_b(x) = 1$ iff $a \leq x < a + 2^m$, and let f_{1-b} be the zero function over $\{0,1\}^n$. The locking oracle will be the combination of the two functions $\sigma = (f_0|f_1)$ (indexed by the first bit of the query to σ).*
2. *Receiver R samples $A \leftarrow \mathcal{A}$ from the family of matrices of Lemma 7 conditioned on the last m rows of A being independent*[3] *and sends A to S.*
3. *Sender S checks that the last m rows of A are independent, and if so he sends $h = aA$ to the receiver R.*

The decommitment phase of Λ_{1R}:

1. *Sender S sends (b, a) to the receiver R.*
2. *Receiver R does the following checks and rejects if any of them does not hold.*
   (a) Check that $aA = h$.
   (b) Check that $f_{1-b}(a) = 0$, and $f_b(a) = 1$.
   (c) For each $i \in [0, m]$, sample $10n$ random points from $[a, a + 2^i)$ and check that $f_b(x) = 1$ for all of them, and also sample $10n$ random points from $(a - 2^i, a - 1]$ and check that $f_b(x) = 0$ for all of them

Proof (of Theorem 6).

Now we study the properties of the ILS Λ_{1R}.

Completeness and Equivocability are immediate.

[3] Note that the last rows of A are independent with probability $1 - 2^{-m} = 1 - 2^{-n}$.

Binding. As a mental experiment we pretend that the randomness used during the decommitment phase by R is chosen in the decommitment phase (rather than in the beginning of the commitment phase).

For a fixed locking oracle σ, Let X_0 (resp. X_1) be the set of possible values of a that sender S can send to the receiver R as the decommitment of $b = 0$ (resp. $b = 1$) and get accepted in the decommitment phase with probability at least 2^{-2n}. We prove that by the end of the commitment phase, with probability at least $1 - 2^{-n/8}$, it holds that $|X_0| = 0$ or $|X_1| = 0$ which means that the sender has only one way to decommit the value b and get accepted with probability more than 2^{-2n}. But now if we choose the receiver's randomness in the commitment phase, since there are at most 2^{n+1} possible values for (b, a), it follows by a simple average argument that with probability at least $1 - 2^{2n-n-1}$ over the commitment phase, the prover gets committed to only one possible value for (b, a) which he can use to pass the decommitment phase successfully.

Claim. $X_0 \cap X_1 = \varnothing$.

Proof. If $a \in X_0 \cap X_1$. Then when a is used as the decommitment of 0, in Step 2b of the decommitment phase the receiver R checks that $f_0(a) = 1, f_1(a) = 0$. On the other hand in the case of decommitting to 1, receiver R checks that $f_b(a) = 0, f_{1-b}(a) = 1$, but they can't both hold at the same time.

Claim. It holds that $|X_0| \leq 2^{n-m}$ and $|X_1| \leq 2^{n-m}$.

Proof. We show that if $\{a, a'\} \subset X_0$ then $|a - a'| \geq 2^m$ (and this would show that $X_0 \leq 2^n/2^m$). Assume on the contrary that $a' < a$ and $a - a' < 2^m$. Let $i \in [1, m]$ be such that $2^{i-1} \leq a - a' < 2^i$. Then by the pigeonhole principle ether at least half of $\sigma([a', a])$ are zero or at least half of the values $\sigma([a', a])$ are one. Without loss of generality let assume that at least half of $\sigma([a', a])$ is zero. In this case at least 1/4 of the values $\sigma([[a', a' + 2^i)])$ are zero. But then by Step 2c of the decommitment phase $(0, a')$ will be accepted with probability at most $(3/4)^{10n} < 2^{-2n}$, and therefore $a' \notin X_0$ which is a contradiction.

Claim. With probability at least $1 - 2^{\Omega(n)}$ over the choice of A, it holds that $|X_0| = 0$ or $|X_1| = 0$.

Proof. Fix any pair $a_0 \in X_0$ and $a_1 \in X_1$, we know that $a_0 \neq a_1$. Therefore, $\Pr_A[a_0 A = a_1 A] = \Pr_A[(a_0 - a_1)A = 0] \leq 2^{-m}$. Claim 3 yields that there are at most $2^{n-m}2^{n-m}$ such pairs, so by using a union bound, with probability at least $1 - 2^{-m}2^{2n-2m} = 1 - 2^{2n-3m}$ over the choice of A, it holds that $X_0 A \cap X_1 A = \varnothing$ which implies that if the sender sends any hash value h, the consistency check of Step 2a of the decommitment phase either makes $|X_0| = 0$ or $|X_1| = 0$.

As we said before Claim 3 implies that with probability $1 - \text{poly}(n) \cdot 2^{2n-3m} = 1 - \text{poly}(n) \cdot 2^{-n/4} \geq 1 - 2^{-n/8}$ over the interaction in the commitment phase the sender gets bound to a fixed $b \in \{0, 1\}$ to which he can decommit successfully.

Hiding. Suppose receiver R can ask at most $u \leq 2^{n/8}$ queries from the locking oracle σ. We claim that before sending the matrix A, all of receiver R's queries to σ are answered zero with probability at least $1-2^{-n/4}$. To see why, think of Z_{2^n} as being divided into $2^{n-m} = 2^{n/4}$ equal intervals such that a is the beginning of one of them. Since receiver R asks up to $2^{n/8}$ queries, before sending the matrix Z, he will ask a query from the interval beginning with a with probability at most $2^{n/8}/2^{n/4} = 2^{-n/8}$. Therefore (up to $2^{-n/8}$ statistical distance in the experiment) we can assume that the matrix A is chosen by receiver R independently of a.

After receiving h, the information that the receiver R knows about a is that it satisfies the equation $aA = h$. If we choose and fix the first $n - m$ bits of (a potential) a, then the remaining bits are determined uniquely because the last m rows of A are full rank. It means that for every $y \in [0, 2^{n-m})$ there is a unique solution for a in the interval $[y2^m, y2^m + 2^m)$, and they are all equally probable to be the true answer from the receiver's point of view.

Now again we claim that (although there are 2^m nonzero points in f_b) all the queries that the receiver R asks from f_b are answered 0 with probability at least $1-2^{-n/8}$. Let $Z = \{z \mid zA = h\}$ be the set of possible values for a. For $z \in Z$, let $I(z) = [z, z+2^m)$. We claim that no $x \in \{0,1\}^n$ can be in $I(z)$ for three different z's from Z. To see why, let $z_1 < z_2 < z_3$ and that $x \in I(z_1) \cap I(z_2) \cap I(z_3)$. But now the interval $[y2^m, y2^m + 2^m)$, containing z_2 separates z_1 and z_3, and so $z_3 - z_1 > 2^m$. Therefore $I(z_1) \cap I(z_3) = \varnothing$ which is a contradiction. So, if the receiver R asks u queries from f_b, he can ask queries from $I(z)$'s for at most $2u$ different z's (out of 2^{n-m} many of them). As a mental experiment assume that a is chosen from Z after the receiver R asked his queries, it holds that $I(a)$ will be an interval that the receiver R never asked any query from with probability at least $1 - u/2^{n-m} \geq 1 - \cdot 2^{-n/8}$. Therefore with probability at least $1 - 2^{-n/9}$ all of receiver R's queries during the commitment phase will be answered zero. But putting the oracle queries aside, the hash value h does not carry any information about the bit-message b and therefore the scheme is $(1 - 2^{n/8})$-hiding.

Now we turn to proving Part 2 of Theorem 6.

By a *noninteractive locking scheme* (NLS), we mean an ILS where the commitment phase is noninteractive and sender S only participates in the decommitment phase. Note that an efficient locking scheme by definition uses poly(n)-sized circuits to implement the locking oracle σ, and therefore σ can have at most poly(n) entropy. In this section we show that there exist no efficient NLS with super-polynomial security.

Since we are going to prove that NLS's cannot be efficient, we need to deal with unbounded senders. Thus we can no longer assume that the decommitment phase is only a message (b, r_S) sent to the receiver, because the randomness r_S used by the sender can be exponentially long. Therefore to prove the strongest possible *negative* result, we allow the decommitment phase of a NLS to be interactive.

The following theorem clearly implies Part 2 of Theorem 6.

Theorem 9. *Let $\Lambda = (S, \sigma, R)$ be any NLS for message space $\{0,1\}$ in which the function σ of the locking oracle has Shannon entropy at most $H(\sigma) \leq \frac{uq}{1000}$ when the committed bit b is chosen at random $b \leftarrow \{0,1\}$. Let u be an upper bound on* *the number of oracle queries to σ asked by the receiver R in the decommitment phase. Then either of the following holds:*

- **Violation of binding:** *There is a fixed locking oracle $\widehat{\sigma}$, and a sender strategy $\widehat{S}$ such that when $\widehat{\sigma}$ is used as the locking orale, for both $b = 0$ and $b = 1$, $\widehat{S}$ can decommit successfully with probability at least $4/5$.*
- **Violation of hiding:** *There exists an unbounded receiver $\widehat{R}$ who can guess the random bit $b \leftarrow \{0, 1\}$ used by (S, σ) with probability at least $4/5$ by asking at most u queries to the locking oracle σ.*

Ideas of the proof of Theorem 9. Our main tool in proving Theorem 9 is the notion of "canonical entropy learner" (EL). Roughly speaking, EL is an efficient-query (computationally unbounded) algorithm which learns a randomized function f (with an oracle access to f) under the uniform distribution assuming that f has a bounded amount of entropy. EL proceeds by choosing to ask one of the "unbiased" queries of f at any step and stop if such queries do not exist. An unbiased query x is one whose answer $f(x)$ is not highly predictable with the current knowledge gathered about f by EL. Whenever EL chooses to ask a query it learns non-negligible entropy of f, and thus the process will stop after $\mathrm{poly}(n)$ steps. On the other hand, when EL stops, all the remaining queries are biased and thus will have a predictable answer *over the randomness of f*. We prove that either the receiver is able to find out the secret message of the sender (in an NLS) by running the EL algorithm, or otherwise if by the end of the learning phase still part of the entropy left in the locking oracle is hiding the secret message, then a malicious prover can plant at least two different messages in the locking oracle in such a way that it can decommit to successfully.

4 On Oblivious Transfer from Stateless Hardware Tokens

In this section we prove that in the stateless hardware token model, there is no statistically secure protocol for oblivious transfer (OT), when the only limitation on malicious parties is being bounded to make polynomially many queries to the tokens.

The stateless token model. In the stateless (tamper-proof hardware) token model, two (computationally unbounded) interactive algorithms A and B will interact with the following extra feature to the standard model. Each party at any time during the protocol can construct a circuit T and put it inside a "token" and send the token T to the other party. The party receiving the token T will have *oracle access* to T and is limited to ask $\mathrm{poly}(n)$ number of queries to the token. The parties can exchange $\mathrm{poly}(n)$ number of tokens during the interaction. The stateless token model clearly extends the IPCP model in which there is only one token sent from the prover to the verifier in the beginning of the game. Therefore proving any *impossibility* result in the stateless token model clearly implies the same result for the the IPCP model. It is easy to see that without loss of generality the parties can avoid sending "explicit messages" to each other and can only use tokens (with messages planted inside the tokens) to simulate all the classical communication with the tokens.

Oblivious transfer by semi-honest parties. If one of the parties is semi-honest (i.e. runs the protocol honestly, and only remember's its view for further off-line investigation), then in fact unconditionally secure OT *is* possible in the stateless token model. If the receiver is honest, then the protocol is simply a token T sent from the sender which encodes $T(0) = x_0, T(1) = x_1$. The receiver will read $T(i)$ to learn x_i. Moreover it is well known that secure OT in one direction implies the existence of secure OT in the other direction, so if the sender is semi-honest unconditionally secure OT is possible in the stateless token model.

We prove that unconditionally secure OT is impossible in the stateless token model, if both parties are *slightly* more malicious than just being semi-honest. Roughly speaking, we define the notion of "curious" parties who run the original protocol (honestly), but will ask more queries from the tokens along the way.[4] We will prove that for any protocol (A, B) aiming to implement OT, there are curious extensions of the original parties $(A_{\mathsf{cur}}, B_{\mathsf{cur}})$ who break the security of the protocol. We prove the following theorem.

Theorem 10 (No unconditional OT from stateless tokens). *Let (S, R) be any protocol for the oblivious transfer in the stateless token model. Then there are curious extensions $(S_{\mathsf{cur}}, R_{\mathsf{cur}})$ to the original algorithms where $(S_{\mathsf{cur}}, R_{\mathsf{cur}})$ (and thus (S, R)) is not a secure protocol for oblivious transfer even when the inputs are random. More formally either of the following holds:*

- **Violation of sender's security:** *When the sender S chooses x_0 and x_1 at random from $\{0, 1\}$ and interacts with R_{cur}, then R_{cur} can find out both of x_0 and x_1 with probability at least* $51/100$*.*
- **Violation of receiver's security:** *When the receiver R chooses $i \leftarrow \{0, 1\}$ at random and interacts with S_{cur}, then S_{cur} can guess i correctly with probability at least* $51/100$*.*

For a high level description of the ideas behind Theorem 10 we refer the reader to the discussion in the Introduction.

Perhaps surprisingly we show that if the parties are allowed to build tokens *around* the tokens received from the other party, then unconditional (UC) secure computation is possible by using *stateless* tokens.

UC secure OT by encapsulation. For a discussion on ideas behind our UC secure OT by token encapsulation we refer the reader to the Introduction and for more details to the full version of the paper.

[4] The term "honest but curious" is sometimes used equivalent to "semi-honest". Our notion is different from both of them because a curious party deviates from the protocol slightly by learning more but emulates the original protocol honestly.

References

1. Rabin, M.O.: How to exchange secrets by oblivious transfer. TR-81, Harvard (1981)
2. Kilian, J.: Founding cryptography on oblivious transfer. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC (1988)
3. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: FOCS, pp. 42–52 (1988)
4. Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. J. Cryptology 5(1), 53–66 (1992)
5. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC, pp. 1–10 (1988)
6. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC, pp. 11–19 (1988)
7. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: STOC, pp. 73–85 (1989)
8. Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: How to remove intractability assumptions. In: STOC, pp. 113–131 (1988)
9. Moran, T., Segev, G.: David and Goliath commitments: UC computation for asymmetric parties using tamper-proof hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 527–544. Springer, Heidelberg (2008)
10. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 308–326. Springer, Heidelberg (2010)
11. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989); Preliminary version in STOC 1985 (1985)
12. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(1), 691–729 (1991); Preliminary version in FOCS 1986 (1986)
13. Fortnow, L.: The complexity of perfect zero-knowledge. Advances in Computing Research: Randomness and Computation 5, 327–343 (1989)
14. Aiello, W., Håstad, J.: Statistical zero-knowledge languages can be recognized in two rounds. J. Comput. Syst. Sci. 42(3), 327–345 (1991)
15. Ostrovsky, R., Wigderson, A.: One-way fuctions are essential for non-trivial zero-knowledge. In: ISTCS, pp. 3–17 (1993)
16. Lapidot, D., Shamir, A.: A one-round, two-prover, zero-knowledge protocol for np. Combinatorica 15(2), 204–214 (1995)
17. Babai, L., Fortnow, L., Lund, C.: Non-deterministic exponential time has two-prover interactive protocols. In: FOCS, pp. 16–25 (1990)
18. Dwork, C., Feige, U., Kilian, J., Naor, M., Safra, S.: Low communication 2-prover zero-knowledge proofs for np. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 215–227. Springer, Heidelberg (1993)
19. Arora, S., Safra, S.: Probabilistic checking of proofs: A new characterization of np. J. ACM 45(1), 70–122 (1998)
20. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. J. ACM 45(3), 501–555 (1998)
21. Fortnow, L., Rompel, J., Sipser, M.: On the power of multi-prover interactive protocols. In: Theoretical Computer Science, pp. 156–161 (1988)

22. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)
23. Kilian, J., Petrank, E., Tardos, G.: Probabilistically checkable proofs with zero knowledge. In: STOC: ACM Symposium on Theory of Computing, STOC (1997)
24. Kalai, Y.T., Raz, R.: Interactive PCP. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 536–547. Springer, Heidelberg (2008)
25. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: interactive proofs for muggles. In: STOC, pp. 113–122 (2008)
26. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way permutation. Journal of Cryptology 11(2), 87–108 (1998); Preliminary version in CRYPTO 1992 (1992)
27. Ostrovsky, R., Venkatesan, R., Yung, M.: Fair games against an all-powerful adversary. In: AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp. 155–169 (1993); Preliminary version in SEQUENCES 1991 (1991)
28. Ding, Y.Z., Harnik, D., Rosen, A., Shaltiel, R.: Constant-round oblivious transfer in the bounded storage model. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 446–472. Springer, Heidelberg (2004)
29. Haitner, I., Reingold, O.: A new interactive hashing theorem. In: IEEE Conference on Computational Complexity, pp. 319–332 (2007); See also preliminary draft of full version at the first author's home page
30. Katz, J.: Universally composable multi-party computation using tamper-proof hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer, Heidelberg (2007)
31. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145 (2001)
32. Goldreich, O., Ostrovsky, R.: Software protection and simulation on oblivious rams. J. ACM 43(3), 431–473 (1996)
33. Goldwasser, S., Kalai, Y.T., Rothblum, G.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)
34. Hazay, C., Lindell, Y.: Constructions of truly practical secure protocols using standardsmartcards. In: ACM Conference on Computer and Communications Security, pp. 491–500 (2008)
35. Chandran, N., Goyal, V., Sahai, A.: New constructions for UC secure computation using tamper-proof hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 545–562. Springer, Heidelberg (2008)
36. Kolesnikov, V.: Truly efficient string oblivious transfer using resettable tamper-proof tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 327–342. Springer, Heidelberg (2010)
37. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
38. Haitner, I., Reingold, O., Vadhan, S.P., Wee, H.: Inaccessible entropy. In: STOC, pp. 611–620 (2009)
39. Kushilevitz, E., Lindell, Y., Rabin, T.: Information-theoretically secure protocols and security under composition. In: STOC: ACM Symposium on Theory of Computing, STOC (2006)
40. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 37–56. Springer, Heidelberg (1990)