On Efficient Zero-Knowledge PCPs

Yuval Ishai[1,⋆], Mohammad Mahmoody[2,⋆⋆], and Amit Sahai[3]

[1] Technion, Israel
yuvali@cs.technion.edu
[2] Cornell, USA
mohammad@cs.cornell.edu
[3] UCLA, USA
sahai@cs.ucla.edu

Abstract. We revisit the question of *Zero-Knowledge PCPs*, studied by Kilian, Petrank, and Tardos (STOC '97). A ZK-PCP is defined similarly to a standard PCP, except that the view of any (possibly malicious) verifier can be efficiently simulated up to a small statistical distance. Kilian et al. obtained a ZK-PCP for **NEXP** in which the proof oracle is in $\mathbf{EXP}^{\mathbf{NP}}$. They also obtained a ZK-PCP for **NP** in which the proof oracle is computable in polynomial-time, but this ZK-PCP is only zero-knowledge against *bounded-query* verifiers who make at most an *a priori fixed* polynomial number of queries. The existence of ZK-PCPs for **NP** with efficient oracles and arbitrary polynomial-time malicious verifiers was left open. This question is motivated by the recent line of work on cryptography using tamper-proof hardware tokens: an efficient ZK-PCP (for any language) is *equivalent* to a statistical zero-knowledge proof using only a single stateless token sent to the verifier.

We obtain the following results regarding efficient ZK-PCPs:

Negative Result on Efficient ZK-PCPs. Assuming that the polynomial time hierarchy does not collapse, we settle the above question in the negative for ZK-PCPs in which the verifier is *nonadaptive* (i.e. the queries only depend on the input and secret randomness but not on the PCP answers).

Simplifying Bounded-Query ZK-PCPs. The bounded-query zero-knowledge PCP of Kilian et al. starts from a *weakly-sound* bounded-query ZK-PCP of Dwork et al. (CRYPTO '92) and amplifies its soundness by introducing and constructing a new primitive called *locking scheme* — an unconditional oracle-based analogue of a commitment scheme. We simplify the ZK-PCP of Kilian et al. by presenting an elementary new construction of locking schemes. Our locking scheme is purely combinatorial.

Black-Box Sublinear ZK Arguments via ZK-PCPs. Kilian used PCPs to construct sublinear-communication zero-knowledge arguments for **NP** which make a *non-black-box* use of collision-resistant hash functions (STOC '92). We show that ZK-PCPs can be used to get black-box variants of this result with improved round complexity,

⋆ Research done in part while visiting UCLA.
⋆⋆ Research done in part while visiting UCLA.

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 151–168, 2012.

as well as an *unconditional* zero-knowledge variant of Micali's non-interactive CS Proofs (FOCS '94) in the Random Oracle Model.

Keywords: Zero-Knowledge, Probabilistically Checkable Proofs, Arthur Merlin Games, Tamper-Proof Tokens, Sublinear Arguments.

1 Introduction

The seminal work of Goldwasser, Micali, and Rackoff [30] changed the classical notion of a mathematical proof by incorporating randomness and interaction. This change was initially motivated by the intriguing possibility of zero knowledge proofs – proofs that carry no extra knowledge other than being convincing. The result of Goldreich, Micali, and Wigderson [27] showed that any **NP** statement can be proved in a zero-knowledge (ZK) manner, making ZK proofs a central tool for cryptographic protocol design; this was later extended by Ben-Or et al. [8] to any language in **PSPACE**. All these fundamental results, however, relied on the assumption that one-way functions exist. Ostrovsky and Wigderson [46] showed that (similar) computational assumptions are indeed inherent for non-trivial zero-knowledge.

Motivated by the goal of achieving *unconditionally* secure zero-knowledge proofs for **NP**, Ben-Or, Goldwasser, Kilian and Wigderson [9] introduced the model of multi-prover interactive proofs (MIP) and presented a perfect ZK protocol for any statement that is provable in the MIP model. Shortly after, Babai, Fortnow, and Lund [6] showed that in fact any language in **NEXP** can be proved in the MIP model. Fortnow, Rompel, and Sipser [23] studied the MIP model further and observed that as a proof system it is equivalent to another model in which an *oracle* encodes a probabilistically checkable proof (PCP) which is queried by an efficient randomized verifier. (The PCP oracle is often identified with the proof string defined by its truth-table, in which case the output domain of the oracle is referred to as the *PCP alphabet*.) The difference between a prover and a PCP oracle is that a prover can keep an internal state, and hence its answer to a given question can depend on other questions. Therefore, soundness against a PCP oracle is potentially easier to achieve than soundness against a malicious prover. This line of work culminated in the celebrated PCP theorem [4,3].

Zero-Knowledge PCPs. In this work we study *zero-knowledge proofs* in the PCP model. A zero-knowledge PCP (ZK-PCP) is defined similarly to a standard PCP, except that the view of any (possibly malicious) verifier can be efficiently simulated up to a small statistical distance. It is instructive to note that zero-knowledge PCPs are incomparable to traditional ZK proofs: since the PCP model makes the prover less powerful, achieving soundness may become easier whereas achieving zero-knowledge may become harder.

The original ZK protocol of [27] for **NP** implicitly relies on *honest-verifier* zero-knowledge PCP for the **NP**-complete problem of 3-coloring of graphs. In this PCP the prover takes any 3-coloring of the input graph, randomly permutes the 3 colors, and writes down the colors as the PCP string. The verifier chooses a random edge,

reads the colors of the vertices of that edge, and accepts iff the colors are different. This ZK-PCP has two disadvantages: **(1)** it is only zero-knowledge against *honest verifiers* (a malicious verifier can learn whether the colors of two non-adjacent nodes are identical), and **(2)** the soundness error is very large: $1 - 1/m$ where m is the number of edges. Dwork et al. [19],[1] relying on the PCP theorem [3,4], improved the ZK-PCP implicit in [27] in both directions. Their construction implies a ZK-PCP for **NP** of polynomial length and with a constant alphabet size such that: **(1)** the PCP is zero-knowledge against verifiers who ask *any* pair of queries (but not more), and **(2)** the soundness error is constant. However, the soundness error of this ZK-PCP could not be easily reduced further while maintaining ZK against malicious verifiers. Furthermore, it could not be made zero-knowledge against arbitrary polynomial-time verifiers, simply because it has polynomial length and a malicious verifier could read the entire proof string.

Kilian, Petrank, and Tardos [40] were the first to explicitly study the power of ZK-PCPs with malicious verifiers. Their work shows how to get around the above limitations, resulting in two kinds of ZK-PCPs with security against malicious verifiers. For the case of languages in **NP**, [40] obtain a PCP of polynomial length over a binary alphabet which is zero-knowledge with negligible soundness error against malicious verifiers who are limited to ask only up to any *fixed* polynomial $p(|x|)$ number of queries, whereas the honest verifier only asks polylog$(|x|)$ queries to verify the PCP. (The length of the PCP string can be polynomially larger than $p(|x|)$.) We call such PCPs *bounded-query* ZK. For the case of languages in **NEXP**, a scaled up version of this construction yields a ZK-PCP in which honest verifiers are efficient (i.e. run in poly$(|x|)$ time), but soundness holds against *arbitrary* polynomial time verifiers. However, the PCP oracle in this case cannot be computed in polynomial time even for languages in **NP**. (By "computable in polynomial time" we mean that the oracle outputs a polynomial-time computable function of its secret randomness, the input x, the **NP**-witness, and the verifier's query.) This is inherent to the approach of [40], as it requires the entropy of the PCP oracle to be bigger than the number of queries made by a malicious verifier.

The above state of affairs leaves open the following natural question.

> **Main Question:** *Are there efficiently computable PCPs for* **NP** *which are statistically zero-knowledge against any polynomial-time verifier?*

An additional motivation to study the question above comes from the recent line of work on cryptography in an extended model of interaction with "tamper-proof hardware tokens" [38,44,14,29,34,41,33]. This model allows the parties to generate and exchange tamper-proof hardware tokens which are simply circuits (with or without internal state) that are accessible only as a black-box. Indeed, an efficient ZK-PCP for **NP** is equivalent to a statistical zero-knowledge proof for **NP** in this model where the only message sent to the verifier is a single *stateless* token. The stateless nature of the PCP oracle (inside the token) would make such a protocol secure against "resetting attacks" [13]. With this motivation in mind, we revisit the feasibility question of efficient ZK-PCPs for **NP**.

[1] This formulation of the result of [19] is due to [40].

2 Our Results

Our main theorem provides a negative answer to the main question above for the case of *nonadaptive* (honest) verifiers whose queries can only depend on their randomness and the input x but not on the prover's answers (so all the queries can be prepared and asked in one round). This theorem may be viewed as supporting the conjecture that efficient ZK-PCPs for **NP** do not exist.

In the setting of bounded-query ZK-PCPs, we revisit the construction of [40] and simplify it considerably. Our contribution is to present a simple combinatorial construction of a "locking schemes" which was the main tool developed in [40] and used in *both* of their constructions for **NP** and **NEXP**.

Finally, motivated by a line of work on the power of black-box constructions in cryptography, we show that efficient bounded-query ZK-PCPs can be used to make the sublinear-communication zero-knowledge argument construction of Kilian [39] *black-box*. Kilian's construction assumes the existence of a collision-resistant hash function, but it uses the hash function in a non-black-box way. We also obtain constant-round variants of this result and an unconditional non-interactive variant in the Random Oracle Model. In the following we describe our results more formally and put them in the proper context

2.1 Efficient Nonadaptive ZK-PCPs

We prove the following negative result about the existence of ZK-PCPs for **NP**.

Theorem 1 (Main Theorem). *If there exists an efficiently computable PCP for* **NP** *with a nonadaptive honest verifier, constant soundness error, and zero-knowledge against arbitrary polynomial-time verifiers, then the polynomial-time hierarchy collapses.*

What we prove is actually more general than the statement of Theorem 1. Namely, we show that any language with an efficient ZK-PCP of polynomial Shannon entropy (see Remark 4) and a nonadaptive verifier is in **coAM**, and Theorem 1 follows by the result of [12]. Also, we only require the zero-knowledge to hold also against nonadaptive verifiers (of arbitrary polynomial time).[2]

We emphasize that even though the zero-knowledge property of ZK-PCPs is defined in a statistical fashion, our main theorem above does *not* follow from the classical result of Fortnow, Aiello, and Håstad [1,22] who proved that $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$. The reason is that although achieving zero-knowledge in the PCP model is harder, achieving soundness in this model is potentially *easier*.[3] Therefore the languages which posses efficient ZK-PCPs (as far as we know) are not necessarily included in **SZK**. Also recall that if one does not require the

[2] The requirement that the honest verifier be nonadaptive is a restriction to our Theorem 1, but only requiring the zero-knowledge to hold against nonadaptive verifiers makes our result stronger.

[3] The latter comparison manifests itself in the following characterizations: it holds that $\mathbf{PCP}(\mathrm{poly}, \mathrm{poly}) = \mathbf{MIP} = \mathbf{NEXP}$ while $\mathbf{IP} = \mathbf{PSPACE} \subseteq \mathbf{EXP}$.

PCP oracle to be efficiently computable, by the result of [40] all of the languages in **NEXP** (including **NP**) *do* have (statistical) ZK-PCPs.

Using Theorem 1 itself, we can extend Theorem 1 to the case of adaptive (honest) verifiers, as long as the total length of the prover's answers returned in an honest PCP verification is $O(\log n)$ bits (see Corollary 7).

Ideas and Tools. At a high level the proof of Theorem 1 uses ideas from many previous influential works [26,1,20,11] and tools from old and new results in the context of constant-round proofs [31,28,36]. The main challenges are in how to force an untrusted prover to extract a PCP oracle from the simulator and run the honest verifier against this PCP. The soundness of this protocol follows from the soundness of the original PCP. To get the completeness, we need to extract this PCP in a way that it is "close" to an actual accepting PCP, and this is where we use efficiency of the PCP and its bounded entropy. Section 3 is dedicated to describing the main result formally and the main ideas behind it. See the full version of the paper for a formal description of our **AM** protocol.

Motivation and Related Work. A recent line of work in cryptography [38,44,14,29,34,41,33] studies the possibility of obtaining secure protocols in an extended model of interaction in which the parties are allowed to exchange more than just classical bits: the parties are allowed to locally construct a (stateful or stateless) circuit, put it inside a tamper-proof token, and send it to another party. The receiver of a token (in this model) is allowed only to use it as a black-box. Namely, she is only allowed to give inputs to the token and receive the output. (If the token is stateful, asking the same query twice might lead to different answers.) Designing protocols in this model is made challenging by the fact that a receiver of a token has no guarantee that the token is indeed well formed. The work of Goyal et al. [34] showed that any two-party functionality (e.g. zero-knowledge proof) can be carried out securely in this model without relying on computational assumptions. Unfortunately the solution of [34] uses *stateful* tokens, which makes it vulnerable to "resetting attacks". Namely, there is no security guarantee if a malicious party receiving a token can reset it to its initial state, say, by cutting off its power.

In another line of research, Kalai and Raz [37] introduced the Interactive PCP (IPCP) model which is a hybrid between the two-prover and the PCP models. In the IPCP model the verifier interacts with a prover and a PCP oracle. Note that when the prover and the PCP oracle are efficiently computable, the IPCP model becomes a special case of the tamper-proof token model in which the prover sends a *stateless* token (computing the PCP) to the verifier.

Although Kalai and Raz [37] introduced the IPCP model for the purpose of optimizing the PCP length at the cost of small amount of interaction with the prover, Goyal, Ishai, Mahmoody, and Sahai [33] showed that the IPCP model is also interesting for cryptographic purposes in the context of achieving unconditional security in the tamper-proof token model. It was shown in [33] that unconditional (statistical) ZK proofs for **NP** exist in the IPCP model, and moreover the prover and the PCP oracle can be implemented efficiently given a witness

w for $x \in L$. The verifier in the protocol of [33] exchanges only four messages with the prover. A main question left open in [33] was whether there exists any protocol that avoids such interaction between the verifier and the prover altogether (i.e. the verifier only interacts with the PCP oracle). It is easy to see that the latter question is equivalent to our main question above! Namely, any positive answer to our main question implies a proof system in which all the communication between the prover and the verifier consists of a single *stateless* token sent to the verifier which hides the circuit computing the PCP oracle and can convince the verifier about the truth of the input statement in a ZK manner.

Therefore, if efficient ZK-PCPs for **NP** exist, they would lead (without any computational assumptions) to "noninteractive" statistical zero-knowledge proofs for **NP** using tamper-proof hardware with the extra feature of being resistant against resetting attacks, since the used token (which computes the PCP oracle) is stateless.

2.2 Simplifying Bounded-Query ZK-PCPs

Our second contribution is a simplification of the ZK-PCP construction of Kilian et al. [40]. The construction of [40] starts from the weakly-sound bounded-query ZK-PCP of [19] and compiles it into a PCP which is zero-knowledge against malicious verifiers of bounded query complexity. The weakly-sound PCP of [19] is zero-knowledge against any k (possibly adaptive) queries, but suffers from the soundness error $1 - 1/\operatorname{poly}(k)$. The main tool introduced and employed in the compiler of [40] is called a "locking scheme", which is an analogue of a commitment scheme in the PCP model. In a locking scheme a sender holds a secret w and randomly encodes it into an oracle σ_w that can be accessed by the receiver R (denoted as R^{σ_w}). The efficient receiver should not be able to learn any information about w through its oracle access to σ_w. On the other hand, the sender can later send a key to the receiver to decommit the value w. The protocol should guarantee that the sender is not able to change his mind about the value w after constructing the oracle σ_w.[4]

Kilian et al. [40] gave an elegant way of using locking schemes to convert a ZK-PCP with $1-1/\operatorname{poly}(k)$ soundness error into a standard ZK-PCP of constant or even negligible error. Unfortunately, the locking scheme of [40] which forms the main technical ingredient of their ZK-PCP constructions is quite complicated to describe and analyze (pages 6 to 12 there) and uses ad-hoc algebraic techniques.

Motivation. Most applications of ZK-PCPs considered in this work either require the stronger unbounded variant (see Section 2.1) or alternatively can rely on an honest-verifier variant (see Section 2.3), which is easier to realize. However, efficient bounded-query ZK-PCPs with security against *malicious* verifiers can also be motivated by natural application scenarios. For instance, one can consider

[4] In other words, a locking scheme can be thought of as a commitment scheme with statistical security guarantees and minimal interaction such that during its commitment phase the sender sends only a single tamper-proof token (containing the oracle σ_w) to the receiver.

the goal of distributing an **NP**-witness among many servers in a way that simultaneously supports a very efficient verification (corresponding to the work of the honest verifier) and secrecy in the presence of a large number of colluding servers (corresponding to the query bound of a malicious verifier). One can also consider a "time-lock zero-knowledge proof" in which a stateless hardware token contains an embedded witness which can be very quickly validated but requires a lot of time to extract. Another motivation behind our simpler locking schemes comes from the line of work aiming at simplifying PCP constructions and making them combinatorial. The main algebraic and technical components in the final PCP construction of Kilian et al. [40] are **(1)** the PCP theorem of [3,4] (which comes in through the construction of [19]) and **(2)** the locking scheme of [40]. The first (more important) component was considerably simplified by Dinur, and here we give a simplified version of the second component. (For a more extensive survey of this line of research see [42] and the references therein.)

In the full version of this paper, we formally present and analyze a simple combinatorial construction of a locking scheme which can be viewed as a noninteractive implementation of Naor's commitment scheme [45] in the PCP model. In the following we describe the main idea.

Technique. We start by reviewing Naor's commitment scheme. In this commitment scheme, the parties have access to a pseudorandom generator $f\colon \{0,1\}^n \mapsto \{0,1\}^{3n}$ and the protocol works as follows:

The receiver chooses a random "shift" $r \xleftarrow{\$} \{0,1\}^{3n}$ and sends it to the sender. The sender, who holds a secret input bit b, chooses a random seed $s \xleftarrow{\$} \{0,1\}^n$ and sends $f(s) + b \cdot r = t$ to the receiver (the addition and multiplication are componentwise over the binary field). In the decommitment phase the sender simply sends (b, s) to the receiver, and the receiver makes sure that $f(s)+b\cdot r = t$ holds to accept the decommitted value.

The binding property holds because the support set of f is of size at most $|f(\{0,1\}^n)| \leq 2^n$, and a random shift $r \xleftarrow{\$} \{0,1\}^{3n}$ with overwhelming probability of at least $1 - 2^n \cdot 2^n \cdot 2^{-3n} = 1 - 2^{-n}$ will have the property that $f(\{0,1\}^n) \cap (f(\{0,1\}^n) + r) = \varnothing$. Thus for such "good" r, by sending t to the receiver the sender will be bound to at most one possible value of b (regardless of the structure of the function f).

On the other hand, the hiding property of the scheme reduces in a *black-box* way to the pseudorandomness of $f(\mathbf{U}_n)$. Namely, if an efficient receiver $\widehat{R}$ can distinguish between $f(s) + r$ and $f(s) + r \cdot b$, another efficient algorithm D who uses $\widehat{R}$ internally is able to distinguish $f(\mathbf{U}_n)$ from a random value $\mathbf{U}_{3n}$. Thus it holds that if the function f is random, the scheme will be *statistically* hiding against receivers who ask at most $\mathrm{poly}(n)$ oracle queries to f. The reason is that a random function f mapping $\{0,1\}^n$ to random values in $\{0,1\}^{3n}$ is statistically indistinguishable from a truly random function as long as the distinguisher is bound to ask at most $2^{o(n)}$ queries to f.

The above observation about the hiding property of Naor's commitment scheme means that if, in the second round of the commitment phase, the sender chooses f to be a truly random function and sends $f(s) + b \cdot r$ to the receiver as well as

(providing oracle access to) $f(\cdot)$, then we get a secure (inefficient) commitment scheme in the *interactive* PCP model without relying on any computational assumption.[5] In our construction of locking schemes we show how to eliminate the first initial message r of the receiver and emulate the role of this shift r by a few more queries asked by the receiver and more structure in the locking oracle.

2.3 Black-Box Sublinear ZK Arguments

Kilian [39], relying on the PCP construction of [5],[6] proved that assuming the existence of exponentially-hard collision-resistant hash functions (CRH) and 2-message statistically-hiding commitments, one can construct a (6-message) statistical ZK argument for **NP** with polylog(n) communication complexity (where n is the input length). Later on, Damgård et al. [17] showed that 2-message statistically-hiding commitments can be obtained from any CRH, which made the existence of exponentially hard CRH sufficient for the construction of Kilian. Micali [43] showed how to make Kilian's protocol noninteractive in the *random oracle model*. The above constructions make a non-black-box use of the underlying collision-resistant hash function.

Our third contribution is to obtain *black-box* constructions of sublinear ZK arguments for **NP** by using bounded-query efficient ZK-PCPs for **NP**. Namely, we observe that the bounded-query ZK-PCP of [19] can be employed to get an alternative to the ZK argument of Kilian [39] for **NP** which uses the underlying CRH function as a black box. (Our protocols are in fact *fully* black-box [49], in the sense that the security reduction makes a black-box use of the adversary, and have black-box simulators.)

Theorem 2 (Black-Box Sublinear ZK Arguments). *Let $\mathcal{H}$ be any family of collision-resistant hash functions. Using $\mathcal{H}$ only as a black-box, one can construct a constant-round ZK argument system for* **NP** *with negligible soundness error and communication complexity sublinear in the witness size. Furthermore:*

- *For the case of an honest verifier, the zero knowledge is statistical, the round complexity is 4 messages, and the protocol is public coin.*
- *For the case of malicious-verifier zero knowledge, the round complexity is 5 messages, and the proof of security requires that the family of CRH be secure against non-uniform adversaries.*
- *If the family of CRH is secure against adversaries running in time $2^{n^{\Omega(1)}}$, then the communication complexity can be made polylogarithmic in the witness size for both honest verifier and malicious verifier settings.*
- *In the random oracle model, there exists an unconditionally secure noninteractive statistical zero knowledge argument system for* **NP** *with negligible soundness error and polylogarithmic communication complexity.*

We prove Theorem 2 in the full version; below we describe the main ideas.

[5] Note that the random oracle $f(\cdot)$ is *not* efficiently computable. The work of [33] presents an *efficient* construction of unconditionally secure commitments in the IPCP model.

[6] The more advanced PCP constructions of [3,4] were not known at that time.

Motivation and Related Work. Our black-box construction of Theorem 2 is motivated by the recent line of work on studying the power of black-box cryptographic constructions vs. that of non-black-box ones (e.g. [24,18,35,15,16,48,50,32]). The goal in this line of work is to understand whether the non-black-box application of an underlying primitive $\mathcal{P}$ which is used in a construction of another (perhaps more complicated) primitive $\mathcal{Q}$ is *necessary* or a black-box construction exists as well. The reason behind studying this question is that the black-box constructions are generally much more efficient (since the source of the non-black-box-ness usually is an extremely inefficient Cook-Levin reduction to an **NP**-complete language). Moreover, black-box constructions are capable of also incorporating any *physical* implementations of the employed primitive $\mathcal{P}$ in the implementation of $\mathcal{Q}$.

Technique. Kilian's argument system, when only required to be sound (and not ZK), has only four messages and uses the hash function as a black-box. The first three messages can be easily made ZK, and it is only the last message from the prover which potentially carries some knowledge. In this last message, the prover reveals some portions of the PCP. To retain the zero-knowledge property, Kilian substitutes the last message (of his 4-message protocol) by a zero-knowledge sub-protocol through which the prover convinces the verifier that he could have revealed the correct portion of the PCP in a way that would cause the verifier to accept. The latter zero-knowledge sub-protocol makes non-black box use of the code of the hash function used in the protocol. Thus, our goal is to remove the zero-knowledge sub-protocol performed at the end.[7]

In order to make Kilian's 6-message ZK argument black-box, we need to know more details about its first 3 rounds. The first message is simply the description of the hash function sent to the prover. Then by using the given hash function and applying a Merkel tree to the PCP the prover hashes down the PCP into a short string which is sent to the verifier as a commitment to whole PCP. With some care, one can make the hash value carry negligible information about the PCP. The third message (from the verifier) consists of the indices of symbols which the PCP verifier chooses to read from the PCP. The prover, in the 4th message reveals the answers to the PCP queries by revealing the relevant paths of the Merkel tree to the verifier. The committed hash value of the PCP (the second message) together with the collision-resistance property of the hash function prevent the prover from changing his mind about the PCP that he committed to in the second message. Thus the soundness of the PCP implies the soundness of the argument system. To keep the last message of this protocol zero-knowledge, as we said, Kilian's prover will *not* simply reveal the relevant preimages, but instead would prove in a zero-knowledge manner, that he knows a set of preimages that would make the PCP verifier accept.

[7] Barak and Goldreich [7] also employ Kilian's approach to get a 4-message universal argument without zero-knowledge. Similarly to Kilian's protocol, to make their protocol zero-knowledge (or just witness indistinguishable) [7] use the hash function in a non-black-box way.

Our main intuitive observation is that if instead of using the PCP of [3,4] one feeds (a direct product version of) the the *bounded-query* ZK-PCP of [19] to the construction of Kilian, then the prover can safely reveal the relevant preimages in the last step of the basic 4-message argument of Kilian and this will not hurt the zero-knowledge property. The key point is that although the employed PCP is zero-knowledge only against bounded-query PCP verifiers, since we are in the prover/verifier setting, the prover can control how many queries of the PCP are read by the verifier, and therefore the bounded-query ZK property of the used PCP will suffice for the argument system to be zero-knowledge. Because our construction is black box, an unconditional result in the random oracle model follows immediately. Since this construction based on collision-resistant hash functions is black-box, it immediately implies an unconditional construction of sublinear ZK arguments in the random oracle model. Using the transformation of [21,43] one can eliminate the interaction using the random oracle and obtain an *unconditional* construction of sublinear ZK arguments for **NP** in the random oracle model. To obtain the result for malicious verifiers (and negligible soundness error), we apply a variant of the Goldreich-Kahan [25] where both prover and verifier use statistically hiding commitments. See the full version of the paper for a formal description of the protocol and its analysis.

Using NIZK? A possible alternative way to get a ZK argument (without using ZK-PCPs) is to use noninteractive zero-knowledge (NIZK) proofs for **NP** [10].[8] To do so, the prover and the verifier should perform a coin-tossing protocol along with the first 3 messages of the basic variant of Kilian's argument system, and this will allow the prover to be able to send a noninteractive zero-knowledge message to the verifier in his last message which proves to the verifier that the prover knows the right preimages of the hash function. This approach benefits from having only 4 messages exchanged, but it still uses the code of the hash function in a non-black-box way, and moreover, one needs to assume the existence of NIZK proofs for NP (*in addition* to the assumption that exponentially-hard collision-resistant hash functions exist).

3 On Nonadaptive Efficient ZK-PCPs

In this section we give a formal statement of Theorem 1 and more details about the intuition behind its proof. See the full version for a complete proof.

Definition 3. *In a* probabilistically checkable proof *(PCP)* $\Pi = (P, V)$ *for a language* L*, the* prover $P = \{\boldsymbol{\pi}_x\}$ *is an (ensemble) of distributions over* proof *oracles,* V *is an efficient* verifier *accessing a proof* $\pi_x \stackrel{\$}{\leftarrow} \boldsymbol{\pi}_x$*, and the following properties hold.*

- **Completeness:** *For every* $x \in L$*, it holds that* $\Pr_{\pi \stackrel{\$}{\leftarrow} \boldsymbol{\pi}_x}[V^{\pi}(x) = 1] \geq 2/3$.
- **Soundness:** *If* $x \notin L$*, then for every oracle* $\widehat{\pi}$ *we have* $\Pr[V^{\widehat{\pi}}(x) = 0] \geq 2/3$.

[8] This variant was pointed out to us by Rafael Pass [47].

The verifier V is nonadaptive *if the queries it asks only depend on its own private randomness and the input x. (A nonadaptive verifier can prepare all of its oracle queries in advance and ask them in one "round".) For the case where $L \in \mathbf{NP}$, a PCP Π is called* efficient *if there is an $\mathbf{NP}$-relation $R_L(x, w)$ associated with L with the following efficiency property. Given any input x and witness w such that $(x, w) \in R_L$, one can efficiently sample a circuit computing a PCP oracle $\pi_x \xleftarrow{\$} \boldsymbol{\pi}_x$.*[9]

Remark 4 (The Entropy of PCPs). For an input $x \in L$, the entropy of the PCP oracle $\boldsymbol{\pi}_x$ is defined similarly to the entropy of any random variable. Note that for a fixed input $x \in L$ (and witness w for $x \in L$, if the PCP is efficient), the distribution of $\boldsymbol{\pi}_x$ is determined by the prover's private randomness. Since there are at most $2^{\mathrm{poly}(k)}$ circuits of size k, any PCP oracle computable by circuits of size at most $k = \mathrm{poly}(n)$ (regardless of whether these circuits are generated efficiently or not) has entropy at most $\log(2^{\mathrm{poly}(k)}) \leq \mathrm{poly}(k) \leq \mathrm{poly}(n)$, simply because any finite random variable $\mathbf{x}$ has Shannon entropy at most $H(\mathbf{x}) \leq \log |\,\mathrm{Supp}(\mathbf{x})|$.

Definition 5. *Let $\Pi = (\{\boldsymbol{\pi}_x\}, V)$ be a PCP for the language L. Π is called (statistical)* zero-knowledge *(ZK) if for every malicious* $\mathrm{poly}(n)$*-time verifier $\widehat{V}$, there is an efficient* simulator Sim *which runs in (expected)* $\mathrm{poly}(n)$*-time and for a sequence of inputs $x \in L$ the output of $\mathsf{Sim}(x)$ is* $\mathrm{neg}(|x|)$*-close to $\mathsf{View}\langle \boldsymbol{\pi}_x, \widehat{V}\rangle(x)$.*[10] *A simulator Sim is called* straight-line *if it uses $\widehat{V}$ only as a black-box and moreover it just outputs the result of a* single interaction *with $\widehat{V}$. Namely, the simulator Sim interacts with $\widehat{V}$ without knowing its secret randomness $r_{\widehat{V}}$, and its output is distributed statistically close to the view of $\widehat{V}^{\boldsymbol{\pi}_x}$.*

Theorem 1 directly follows from Remark 4 and Theorem 6 below.

Theorem 6. *Let $\Pi = (\{\boldsymbol{\pi}_x\}, V)$ be a ZK-PCP for a language L with a nonadaptive verifier V. If (for every fixed input x) the PCP oracle $\{\boldsymbol{\pi}_x\}$ has entropy at most* $\mathrm{poly}(|x|)$*, then $L \in \mathbf{AM} \cap \mathbf{coAM}$. Moreover $L \in \mathbf{BPP}$ if the simulator is straight-line.*[11]

Corollary 7. *Let $\Pi = (\{\boldsymbol{\pi}_x\}, V)$ be a ZK-PCP for a language L with oracle entropy at most* $\mathrm{poly}(n)$*, and suppose the total length of the PCP answers returned to the verifier during a single verification is at most $O(\log n)$ bits, then (regardless of the adaptivity of the verifier), it holds that $L \in \mathbf{AM} \cap \mathbf{coAM}$. (Also $L \in \mathbf{BPP}$ if the simulator is straight-line.)*

[9] More formally, in that case we shall index the oracle distributions $\{\boldsymbol{\pi}_{x,w}\}$ by both the input and the witness. Then the completeness should hold for all $x \in L$ when the prover uses *any* witness w that $x \in L$.

[10] In the case of efficient ZK-PCPs, the zero-knowledge property should hold regardless of which witness w (for $x \in L$) is used by the prover to generate the oracle.

[11] Bounded-query ZK-PCPs of [40] and its predecessors [27,19] all have straight-line simulators.

Note that in Corollary 7 there is no bound on the length of the *queries* of the verifier, and particularly it can be applied to cases that the number of queries of V is $O(\log n)$ and the PCP answers (alphabet) are of constant size while the length of the PCP is exponential $2^{\text{poly}(n)}$ (which makes the length of the queries of the verifier at least $\text{poly}(n)$).

Proof (Proof of Corollary 7). Since the total length of oracle answers is $O(\log n)$ bits, we can modify the verifier V into another equivalent verifier V' as follows: the new verifier V' tries to ask a superset of the queries that V would ask, but V' asks its queries in a nonadaptive way. In particular V' enumerates all the possible answers that V might get from the oracle, continues the verification in each case, and prepares all the possible V queries at the beginning. There are at most $2^{O(\log n)} = \text{poly}(n)$ many possibilities caused by different PCP answers in a verification, thus there will be at most $\text{poly}(n)$ many queries asked by V'. After getting the answers, V' can emulate V internally and decide as V would. The completeness, soundness, and zero-knowledge of V' are inherited from those of V by definition.

3.1 Main Ideas and Framework

Here we describe the main ideas behind the proof of Theorem 6. Our **AM** protocols for $\overline{L}$ and L follow the same general framework. (The **AM** protocol for $\overline{L}$ is the more interesting case, since it implies the collapse of the hierarchy in case L is **NP**.)

First we show that if a bounded-entropy ZK-PCP for L has a straight-line simulator, then L (and $\overline{L}$) can be decided by an efficient **BPP** algorithm D_L. At a very high level, this step uses ideas from [26] by looking at a particular malicious verifier (in our case a repeated version of the honest verifier) and using its interaction with the straight-line simulator to decide the language. Since the key ideas already appear in the case of straight-line simulation, in Section 3.2 below we start by only describing this basic case.

Beyond Straight-Line Simulation. For the case of general (statistical) simulation, we show how to emulate the efficient algorithm D_L above with the help of an untrusted prover. In particular, we first show how to emulate D_L with the help of some advice α_x sampled from a specific distribution[12], and then we will show how to get this advice α_x from an (untrusted) prover through a constant round protocol GetAdv. The latter protocols are implemented following similar frameworks introduced by Feigenbaum and Fortnow [20] (and extended in the followup works of [11,2]) in the context of studying the possibility of worst-case to average-case reductions for **NP**. Our protocol, however, is more complicated and uses recent and old sampling protocols from [31,28,36].

[12] Here we are using the term "advice" in a nonstandard way, because the advice distribution α_x depends on the input x (rather than only depending on the input length $|x|$).

3.2 The Case of Straight-Line Simulation

In this section we present the **BPP** algorithm for L assuming that the ZK-PCP has a perfect straight-line simulator. This special case already captures the main ideas, and we refer the reader to the full version for the general case.

Since the PCP verifier V is assumed to be nonadaptive, we can assume w.l.o.g. that V permutes its queries $a_1, \ldots, a_q$ randomly before querying the oracle.

The Intuition. The general framework is to use the simulator Sim to find a "good enough" *oracle* φ and run a fresh instance of the verifier V against this oracle. This way, the correctness of our algorithm to decide membership in L follows from the soundness of the original PCP system. The challenge is to sample the oracle φ in a way that makes the verifier accept in case the input x is in L. Suppose we run the simulator over the "mildly malicious" verifier who only repeats several (independent) executions of the verifier: $(V^1, \ldots, V^k)$. Then, in case $x \in L$, the simulated transcript of all of these executions $(V^1, \ldots, V^k)$ will be accepted. To define the oracle φ, relying on the straight-line nature of the simulator, we can fix any simulated partial transcript for $(V^1, \ldots, V^i)$ (for $i \in [k]$) and ask Sim to answer any new query *only conditioned* on the simulated transcript of $(V^1, \ldots, V^i)$. (Even though φ is a randomized oracle, its randomness can be fixed independently of the final verification that is executed over φ.) The main intuition is that since the entropy of the simulated transcript for $(V^1, \ldots, V^k)$ is bounded, for most of $i \in [k]$ the simulated transcript of V^i has very small entropy, and relying on the non-adaptivity of V, all of its queries could be thought of as the "first query", and this way the oracle φ (defined above) behaves very close to the actual "oracle" of the simulated transcript of V^i which leads to an accept. The formal argument follows.

Notation. Let $V^{[k]}$ be an execution of k independent copies of the PCP verifier V. By V^i we refer to the i-th execution of V in $V^{[k]}$ (i.e. $V^{[i]} = (V^1, \ldots, V^i)$). $V^{[k]}$ is a potentially malicious verifier whose view $\mathsf{View}\langle \boldsymbol{\pi}_x, V^{[k]} \rangle$ is assumed to be perfectly simulated by the straight-line simulator Sim (when given access to $V^{[k]}$). The view $\mathsf{View}\langle \boldsymbol{\pi}_x, V^{[k]} \rangle$ is composed of k random seeds $r^1, \ldots, r^k$ for V and k transcripts $\tau^1, \ldots, \tau^k$ such that each $\tau^i = (a_1^i, b_1^i, \ldots, a_q^i, b_q^i)$ is a partial transcript where $\{a_1^i, \ldots, a_q^i\}$ are the queries asked by V using the randomness r^i and $b_j^i = \pi_x(a_j^i)$ is (supposedly) a corresponding returned oracle answer. We will only use the fact that Sim simulates $(\tau^1, \ldots, \tau^k)$ correctly and will ignore the fact that this is simulated jointly with random seeds $(r^1, \ldots, r^k)$. Also since we will use Sim only over $V^{[k]}$ and some input x, for simplicity in the following we will use Sim to denote $\mathsf{Sim}(V^{[k]}, x)$. Also, let $m = \mathrm{poly}(n) \geq H(\boldsymbol{\pi}_y)$ be the upper bound on the PCP entropy for every $y \in L \cap \{0,1\}^n$, and let $\epsilon = 1/\mathrm{poly}(n)$ be a parameter controlling the error of the **BPP** algorithm D_L. The formal description of the algorithm D_L is as follows.

Construction 8. BPP Algorithm D_L. *Set $k = m \cdot (\frac{3q}{\epsilon})^2$ where q is the query complexity of V and ϵ is the error parameter.*

1. *Randomly choose* $i \xleftarrow{\$} [k]$, *and use* Sim *to generate* $(\tau^1, \ldots, \tau^{i-1})$ *as prefix of* $\mathsf{View}\langle \boldsymbol{\pi}_x, V^{[i-1]} \rangle$.
2. *Choose a fresh randomness* r^i *for the verifier* V *and generate the queries* $a^i_1, \ldots, a^i_q$ *using* r^i.
3. *Using the simulator* Sim *answer each of the queries* a^i_j *as follows to get the answer* b_j. *We extend the execution of the straight-line simulator* Sim *assuming that* a^i_j *is the* first *query of* V^i *conditioned on* $(\tau^1, \ldots, \tau^{i-1})$ *being generated already for* $(V^1, \ldots, V^{i-1})$.
4. *Finally output whatever* V *decides over the view* $(r^i, a^i_1, b_1, \ldots, a^i_q, b_q)$.

Lemma 9. *If* Π *has soundness* $1 - \delta_s$, *then* D_L *will reject every* $x \notin L$ *with probability* $\geq 1 - \delta_s$, *and if* Π *has completeness* $1 - \delta_c$, *then* D_L *will accept every* $x \in L$ *with probability* $\geq 1 - (\delta_c + \epsilon)$.

Proof (Proof of Lemma 9). We study the cases $x \in \overline{L}$ and $x \in L$ separately.

When $x \in \overline{L}$. The final verification of the algorithm of Construction 8 is run against a *randomized* oracle, but this oracle can be sampled and fixed independently of the randomness of the verifier, thus the soundness of the PCP implies the soundness of D_L. More formally, define the randomized oracle $\boldsymbol{\varphi}^i = (\boldsymbol{\pi}_x \mid \tau^1, \ldots, \tau^{i-1})$ according to the distribution of the PCP oracle $\boldsymbol{\pi}_x$ conditioned on the view of $V^{[i-1]}$. Define the oracle $\widehat{\boldsymbol{\varphi}}^i$ as a randomized oracle that for every new query a it samples a fresh instance of the oracle $\varphi \xleftarrow{\$} \boldsymbol{\varphi}^i$ and then answers a using φ. Based on Construction 8 D_L is indeed running the verifier V against an instance of the oracle $\widehat{\varphi} \leftarrow \widehat{\boldsymbol{\varphi}}^i$ and outputs $V^{\widehat{\varphi}}(x)$. Thus, since $x \notin L$, by the soundness of V, with probability at least $1 - \delta_s$ it holds that $V^{\widehat{\varphi}}(x) = 0$. Note that if instead of asking all of the queries of the verifier "as the first query" we simply ask the simulator to simulate the whole view, the answers might *not* be chosen according to any fixed oracle *independently* of the randomness of V, and V might accept even though $x \in \overline{L}$.

When $x \in L$. Informally speaking, the verifier accepts in this case for the following two reasons: **(1)** If we sample the view of the final verification simply as the view of V^i as an extension of $V^{[i-1]}$ all sampled by the simulator Sim (i.e. using the oracle $\boldsymbol{\varphi}^i$ rather than $\widehat{\boldsymbol{\varphi}}^i$), then it will be an accepted view by the definition of the simulator, moreover **(2)** since the verifier is nonadaptive and permutes its answers, any of its queries can be thought of as the first query. More formally, consider the following two mental experiments:

1. Sample $(\tau^1, \ldots, \tau^{i-1})$ and $\varphi \xleftarrow{\$} \boldsymbol{\varphi}^i$ (as defined above) and sample $a^i_1, \ldots, a^i_q$ (by sampling r^i). Then execute q versions of the verifier V as follows. In the j'th execution ask the queries from φ in this order: $(a^i_j, \ldots, a^i_q, a^i_1, \ldots, a^i_{j-1})$ and receive the answers $(b^i_j, \ldots, b^i_q, b^i_1, \ldots, b^i_{j-1})$.

2. Do the same as above, but here in the j'th execution first sample a *fresh* oracle $\varphi_j \xleftarrow{\$} \boldsymbol{\varphi}^i$ and then ask the queries in the order $(a^i_j, a^i_{j+1}, \ldots, a^i_q, a^i_1, \ldots, a^i_{j-1})$ to get the answers $(c^j_1, \ldots, c^j_q)$.

Claim. Let $\alpha = m/k$. Then for every $j \in [q]$, it holds that $\Pr[b^i_j = c^j_1] \geq 1 - 3\sqrt{\alpha}$.

Now we prove Claim 3.2. A crucial point is that the queries of V are already permuted randomly, and therefore rotations inside each execution will still produce a random execution of V (although these random executions are correlated). Therefore by symmetry, it would suffice to prove Claim 3.2 only for the first execution of the two experiments. Since $H(\boldsymbol{\pi}_x) \leq m$ and that $\mathbf{a}^i_j$'s are sampled independently of $\boldsymbol{\pi}_x$, therefore:

$$m \geq H(\boldsymbol{\pi}_x) \geq \sum_{i \in [k]} \sum_{j \in [q]} H(\mathbf{b}^i_j \mid \mathbf{a}^1_1, \mathbf{b}^1_1, \ldots, \mathbf{a}^i_j) \geq \sum_{i \in [k]} H(\mathbf{b}^i_1 \mid \boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^{i-1}, \mathbf{a}^i_1).$$

By averaging over i and using the definition of the conditional entropy it holds that:

$\mathbb{E}_{i \xleftarrow{\$} [k], \boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^{i-1}, \mathbf{a}^i_1} H(\mathbf{b}^i_1 \mid \boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^{i-1}, \mathbf{a}^i_1) \leq m/k = \alpha$. By another averaging argument, with probability at least $1 - \sqrt{\alpha}$ over sampling and fixing $(i \xleftarrow{\$} [k], \tau^1, \ldots, \tau^{i-1}, a^i_1)$, it would hold that $H(\mathbf{b}^i_1 \mid \tau^1, \ldots, \tau^{i-1}, a^i_1) \leq \sqrt{\alpha}$. We use the following lemma to bound the collision probability when the Shannon entropy is small.

Lemma 10. *For every finite random variable* $\mathbf{x}$ *it holds that* $\Pr_{x_1 \xleftarrow{\$} \mathbf{x}, x_2 \xleftarrow{\$} \mathbf{x}}[x_1 = x_2] \geq 1 - 1.45 H(\mathbf{x})$.

Proof. Let $C = \Pr_{x_1 \xleftarrow{\$} \mathbf{x}, x_2 \xleftarrow{\$} \mathbf{x}}[x_1 = x_2]$ be the collision probability of $\mathbf{x}$, let $p_i = \Pr[\mathbf{x} = i]$, and let $H = H(\mathbf{x})$. By Jensen's inequality: $\sum_i p_i \log p_i \leq \log \sum_i p_i^2$ it holds that $H \geq \log 1/C$ (where $\log 1/C$ is also known as the Renyi entropy). Therefore using $e^{-x} \geq 1 - x$ we conclude that: $C \geq 2^{-H} = e^{(-\ln 2) \cdot H} \geq 1 - (\ln 2) \cdot H > 1 - 1.45H$.

By Lemma 10, the bounded entropy of $H(\mathbf{b}^i_1 \mid \tau^1, \ldots, \tau^{i-1}, a^i_1) \leq \sqrt{\alpha}$ implies that its collision probability is at least $1 - 2\sqrt{\alpha}$ and since $\mathbf{c}^1_1$ and $\mathbf{b}^i_1$ are both sampled from $(\mathbf{b}^i_1 \mid \tau^1, \ldots, \tau^{i-1}, a^i_1)$, we have $\Pr[\mathbf{c}^1_1 = \mathbf{b}^i_1] \geq 1 - 2\sqrt{\alpha}$. Claim 3.2 now follows by a union bound.

Claim 3.2 implies that the sampled $(r^i, a^i_1, b_1, \ldots, a^i_q, b_q)$ in the algorithm D_L (which is the same as using the first query/answer pairs of executions in the second experiment) will also lead to accepting with probability at least $1 - \delta_c - 3q\sqrt{\alpha} = 1 - (\delta_c + \epsilon)$.

Acknowledgement. We thank Vipul Goyal for collaboration at an early stage of this work. We would also like to thank Kai-Min Chung and Rafael Pass for very insightful discussions and the anonymous reviewers for their valuable comments. Yuval Ishai was supported by ERC Starting Grant 259426, ISF grant

1361/10, and BSF grant 2008411. Mohammad Mahmoody was supported in part by NSF Award CCF-0746990, AFOSR Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. Amit Sahai was supported in part by a DARPA/ONR PROCEED award, NSF grants 1136174, 1118096, 1065276, 0916574 and 0830803, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0389. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, Department of Defense, or the US government.

References

1. Aiello, W., Håstad, J.: Statistical zero-knowledge languages can be recognized in two rounds. Journal of Computer and System Sciences 42(3), 327–345 (1991); Preliminary version in FOCS 1987
2. Akavia, A., Goldreich, O., Goldwasser, S., Moshkovitz, D.: On basing one-way functions on np-hardness. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC), pp. 701–710 (2006)
3. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. Journal of the ACM 45(3), 501–555 (1998); Preliminary version in FOCS 1992
4. Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. Journal of the ACM 45(1), 70–122 (1998); Preliminary version in FOCS 1992
5. Babai, Fortnow, Levin, Szegedy: Checking computations in polylogarithmic time. In: STOC: ACM Symposium on Theory of Computing (STOC) (1991)
6. Babai, L., Fortnow, L., Lund, C.: Non-deterministic exponential time has two-prover interactive protocols. In: FOCS, pp. 16–25 (1990)
7. Barak, B., Goldreich, O.: Universal arguments and their applications, pp. 194–203 (2002)
8. 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)
9. Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: How to remove intractability assumptions. In: STOC, pp. 113–131 (1988)
10. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC), pp. 103–112 (1988)
11. Bogdanov, A., Trevisan, L.: On worst-case to average-case reductions for np problems. SIAM Journal on Computing 36(4), 1119–1159 (2006)
12. Boppana, R.B., Håstad, J., Zachos, S.: Does co-NP have short interactive proofs? Information Processing Letters 25, 127–132 (1987)
13. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)
14. 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)

15. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Black-Box Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 427–444. Springer, Heidelberg (2008)
16. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, Black-Box Constructions of Adaptively Secure Protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 387–402. Springer, Heidelberg (2009)
17. Damgard, Pedersen, Pfitzmann: On the existence of statistically hiding bit commitment schemes and fail-stop signatures. Journal of Cryptology 10 (1997)
18. Damgård, I., Ishai, Y.: Constant-Round Multiparty Computation Using a Black-Box Pseudorandom Generator. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 378–394. Springer, Heidelberg (2005)
19. Dwork, C., Feige, U., Kilian, J., Naor, M., Safra, M.: 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)
20. Feigenbaum, J., Fortnow, L.: Random-self-reducibility of complete sets. SIAM Journal on Computing 22(5), 994–1005 (1993)
21. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
22. Fortnow, L.: The complexity of perfect zero-knowledge. Advances in Computing Research: Randomness and Computation 5, 327–343 (1989)
23. Fortnow, L., Rompel, J., Sipser, M.: On the power of multi-prover interactive protocols. Theoretical Computer Science 134(2), 545–557 (1994)
24. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (2000)
25. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology 9(3), 167–190 (1996)
26. Goldreich, O., Krawczyk, H.: On the Composition of Zero-Knowledge Proof Systems. SIAM Journal on Computing 25(1), 169–192 (1996); In: Paterson, M. (ed.) ICALP 1990. LNCS, vol. 443, pp. 268–282. Springer, Heidelberg (1990);
27. 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
28. Goldreich, O., Vadhan, S., Wigderson, A.: On Interactive Proofs with a Laconic Prover. In: Yu, Y., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 334–345. Springer, Heidelberg (2001)
29. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-Time Programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)
30. 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
31. Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. Advances in Computing Research: Randomness and Computation 5, 73–90 (1989)
32. Goyal, V.: Constant round non-malleable protocols using one way functions. In: Fortnow, L., Vadhan, S.P. (eds.) STOC, pp. 695–704. ACM (2011)
33. Goyal, V., Ishai, Y., Mahmoody, M., Sahai, A.: Interactive Locking, Zero-Knowledge PCPs, and Unconditional Cryptography. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 173–190. Springer, Heidelberg (2010)

34. 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)
35. Haitner, I., Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions of protocols for secure computation. SIAM J. Comput. 40(2), 225–266 (2011)
36. Haitner, I., Mahmoody, M., Xiao, D.: A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In: IEEE Conference on Computational Complexity, pp. 76–87. IEEE Computer Society (2010)
37. 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)
38. 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)
39. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC), pp. 723–732 (1992)
40. Kilian, J., Petrank, E., Tardos, G.: Probabilistically checkable proofs with zero knowledge. In: STOC: ACM Symposium on Theory of Computing, STOC (1997)
41. 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)
42. Meir, O.: Combinatorial PCPs with efficient verifiers. In: FOCS, pp. 463–471. IEEE Computer Society (2009)
43. Micali, S.: Computationally sound proofs. SIAM Journal on Computing 30(4), 1253–1298 (2000); Preliminary version in FOCS 1994
44. 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)
45. Naor, M.: Bit Commitment Using Pseudo-Randomness. Journal of Cryptology 4(2), 151–158 (1991); In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 128–136. Springer, Heidelberg (1990)
46. Ostrovsky, R., Wigderson, A.: One-way functions are essential for non-trivial zero-knowledge. In: Proceedings of the 2nd Israel Symposium on Theory of Computing Systems, pp. 3–17. IEEE Computer Society (1993)
47. Pass, R.: Personal communication
48. Pass, R., Wee, H.: Black-Box Constructions of Two-Party Protocols from One-Way Functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (2009)
49. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of Reducibility between Cryptographic Primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004)
50. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: FOCS, pp. 531–540. IEEE Computer Society (2010)