On the Round Complexity of Zero-Knowledge Proofs Based on One-Way Permutations

S. Dov Gordon[1], Hoeteck Wee[2] *, David Xiao[3] **, and Arkady Yerukhimovich[1]

[1] Dept. of Computer Science, University of Maryland, College Park MD 20742, USA
[2] Dept. of Computer Science, Queens College, CUNY, Flushing NY 11367, USA
[3] LRI, Université Paris-Sud, 91405 Orsay Cedex, France

Abstract. We consider the following problem: can we construct constant-round zero-knowledge proofs (with negligible soundness) for **NP** assuming only the existence of one-way permutations? We answer the question in the negative for fully black-box constructions (using only black-box access to both the underlying primitive and the cheating verifier) that satisfy a natural restriction on the "adaptivity" of the simulator's queries. Specifically, we show that only languages in **coAM** have constant-round zero-knowledge proofs of this kind. We also give strong evidence that we are unlikely to find fully black-box constructions of constant-round zero knowledge proofs for **NP**, even without this restriction on adaptivity.

Keywords: constant-round zero-knowledge proofs, black-box separations

* Part of this research was completed while a post-doc at Columbia University. Supported in part by NSF CAREER Award CNS-0953626.

** Part of this reseach was completed while a student at Princeton University.

1 Introduction

A *zero-knowledge proof* is a protocol wherein one party, the prover, convinces another party, the verifier, of the validity of an assertion while revealing no additional knowledge. Introduced by Goldwasser, Micali and Rackoff in the 1980s [18], zero-knowledge proofs have played a central role in the design and study of cryptographic protocols. In these applications, the main measure of efficiency is the *round complexity* of the proof system, and it is important to construct constant-round zero-knowledge protocols for **NP** (with negligible soundness) under minimal assumptions. In many cases, a computational zero-knowledge argument system (where both the zero-knowledge and soundness guarantees hold against computationally bounded adversaries) suffices, and we know how to construct such protocols for **NP** under the minimal assumption of one-way functions [9, 29]. However, in this work, we focus on computational zero-knowledge proof systems, where the soundness guarantee must hold against computationally unbounded adversaries.

A common intuition in constructing zero knowledge protocols (typically based on some form of commitments) is that statistical (resp. computational) soundness corresponds to using a statistically (resp. computationally) binding commitment, while statistical (resp. computational) zero knowledge corresponds to using statistically (computationally) hiding commitments. One might also expect that the round complexity of the resulting zero knowledge protocol is roughly the same as the round complexity of the underlying commitment scheme.

However, the best known construction of computational zero-knowledge proofs from one-way permutations has $\omega(1)$ rounds [16, 7], and the minimal assumption from which we know how to construct constant-round computational zero-knowledge proofs for **NP** is constant-round statistically *hiding* commitments [14, 35], which seem to be a stronger assumption than one-way permutations [37, 21]. There are no known constructions of constant-round computational zero knowledge proofs from constant-round statistically *binding* commitments. We note that the latter may be constructed from one-way permutations [7] and one-way functions [28, 24]. This raises the following intriguing open problem:

> Can we base constant-round zero-knowledge proofs for **NP** on the existence of one-way permutations?

We briefly survey what's known in this regard for constant-round black-box zero-knowledge protocols (that is, those using a black-box simulation strategy). We clarify that while we do know of non-black-box zero-knowledge protocols [2, 20], these protocols are all zero-knowledge arguments and not proofs.

Unconditional Constructions. The only languages currently known to have constant-round zero-knowledge proofs from assumptions weaker than statistically hiding commitment schemes are those that admit statistical zero-knowledge proofs, which do not require any computational assumption at all. Even though this includes languages believed to be outside of **BPP** such as graph isomorphism and graph non-isomorphism [16, 6], all languages with statistical zero knowledge proofs lie in $\mathbf{AM} \cap \mathbf{coAM}$ [1, 11] (and therefore do not include all of **NP** unless the polynomial hierarchy collapses).

Lower Bounds. Goldreich and Krawczyk [15] showed that 3-round zero-knowledge protocols and public-coin constant-round zero-knowledge protocols with black-box simulators exist only for languages in **BPP**. Katz [26] showed that 4-round zero-knowledge proofs only exist for languages in $\mathbf{MA} \cap \mathbf{coMA}$. Haitner et al. [21] ruled out fully black-box constructions of constant-round statistically hiding commitment schemes (in fact, any $O(n/\log n)$-round protocol) from one-way permutations, which means that we are unlikely to obtain constant-round zero-knowledge proofs from one-way permutations via the approach in [14]. More recently, Haitner et al. [23] established a partial converse to [14], namely that any constant-round zero-knowledge proof for **NP** that remains zero-knowledge under parallel composition implies the existence of constant-round statistically hiding commitments. Unlike the case for stand-alone zero-knowledge, we do not know if there exists a $\omega(1)$-round zero-knowledge proof system for **NP** that remains zero-knowledge under parallel composition, assuming only the existence of one-way permutations. Indeed, zero-knowledge under parallel composition appears to be a qualitively much stronger security guarantee than stand-alone zero-knowledge.

1.1 Our Result.

In this work, we establish new barriers towards constructing zero-knowledge proof systems from one-way permutations for all of **NP**:

> **Main Theorem (informal).** Only languages in $\mathbf{AM} \cap \mathbf{coAM}$ admit a fully black-box construction of zero-knowledge proofs starting from one-way permutations where the construction relies on a black-box simulation strategy with constant adaptivity.

A fully black-box construction (c.f. [34, 25]) is one that not only relies on a black-box simulation strategy, but where the protocol relies on black-box access to the underlying primitive. Adaptivity is a measure of how much the black-box simulator relies on responses from previous queries to the cheating verifier in order to generate new queries. We point out that all known constructions of black-box simulators achieve adaptivity that is linear in the round complexity of the protocol and therefore constant adaptivity is a fairly natural restriction for constant-round protocols. Apart from the restriction on adaptivity, this is essentially the best one could hope for in lieu of various positive results mentioned earlier:

- Our result only applies to constant-round protocols – running the $O(\log n)$-fold parallel repetition of Blum's Hamiltonicity protocol [7] sequentially yields a $\omega(1)$-round black-box zero-knowledge proof system for **NP**.

- Our result applies only to proofs, but not arguments – there exists a fully black-box construction of constant-round computational zero-knowledge arguments with constant adaptivity from one-way functions for all of **NP**. [10, 32].
- We have unconditional constructions of constant-round statistical black-box zero-knowledge proofs for graph isomorphism and graph non-isomorphism, languages which are in **AM**$\cap$**coAM** but are commonly believed to lie outside **BPP**.

Limitations of Our Impossibilty Result. Our impossibilty result imposes three main restrictions on the construction: black-box simulation strategy, black-box access to the one-way permutation, and bounded adaptivity of the black-box simulator, amongst which adaptivity appears to be the greatest limitation. Our current ability to prove general lower bounds for zero-knowledge (without limitation to black-box simulation) is relatively limited [17, 4]; moreover, non-black-box simulation strategies so far only yield arguments and not proof systems. In the context of zero-knowledge protocols, there is no indication whether non-black-box access to the underlying primitive has an advantage over black-box access to the primitive.

Extensions to Higher Adaptivity. The formal statement of our result (Theorem 2) is slightly more general than stated above and, in particular, allows us to obtain non-trivial consequences even when the simulator's adaptivity is polynomial.

> **Generalized Main Theorem (informal).** If a language L admits a fully black-box construction of zero-knowledge proofs starting from one-way permutations where the construction relies on a black-box simulation strategy with adaptivity t, then both L and $\overline{L}$ have $O(t)$-round public coin interactive proofs where the honest prover strategy can be implemented in $\mathbf{BPP}^{\mathbf{NP}}$.

For the case $t = O(1)$ this is just our main theorem. If we now let L be an **NP**-complete language, then for $t = O(\log n)$ this implies a collapse in the *quasi-polynomial hierarchy* [33], which one can view as a weakened version of a collapse in the polynomial hierarchy. For $t = o(n)$ this would improve on the best known round complexity for an interactive protocol for a **coNP**-complete language (the best known is linear [27]), and even for $t = \text{poly}(n)$ this would improve on the best known honest prover complexity for an interactive protocol for a **coNP**-complete language (the best known is $\mathbf{P}^{\#\mathbf{P}}$ [27]). We view these results as evidence that such constructions will be hard to find.

1.2 Proof Overview

Recall that we start out with a constant-round zero-knowledge proof system $(\mathcal{P}, \mathcal{V})$ with constant adaptivity for a language L and we want to show that L lies in **AM** $\cap$ **coAM**. The high level strategy is to extend the Goldreich-Krawczyk lower bound for constant-round public-coin protocols [15] to the private-coin setting. Following [15] (also [30, 26, 23]), we consider a cheating verifier $\mathcal{V}^*_{\mathsf{GK}}$ that "resamples" new messages that are distributed identically to the real verifier's messages (conditioned upon the partial transcript) every time it is rewound. We will need to address the fact that we do not know how to simulate such a $\mathcal{V}^*_{\mathsf{GK}}$ efficiently for general private-coin protocols. The computational complexity of $\mathcal{V}^*_{\mathsf{GK}}$ comes up in two ways in [15]: first to deduce that the zero-knowledge property holds against such a $\mathcal{V}^*_{\mathsf{GK}}$, and second to derive an efficient $\mathbf{AM}$ protocol for the underlying language L and its complement $\overline{L}$.

To address the first issue, we rely on a result of Haitner et al. [21], which, roughly speaking, demonstrates the existence of a one-way permutation π secure in the presence of a $\mathcal{V}^*_{\mathsf{GK}}$ oracle (as long as the zero-knowledge protocol has bounded round complexity, which is the case here). We will then instantiate the zero-knowledge protocol $(\mathcal{P}, \mathcal{V})$ with the permutation π. This will remain zero-knowledge against the cheating verifier $\mathcal{V}^*_{\mathsf{GK}}$ since π is one-way against $\mathcal{V}^*_{\mathsf{GK}}$. Following [15, 26, 23], we may then deduce a $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$ algorithm for L. (Such a statement was obtained independently by Pass and Venkitasubramaniam [31].) Along the way, we will exploit (as with [26, 23]) the fact that $(\mathcal{P}, \mathcal{V})$ is a proof system as we need soundness to hold against a cheating prover that is able to simulate $\mathcal{V}^*_{\mathsf{GK}}$.

Next, we will essentially show that $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}} \subseteq \mathbf{AM} \cap \mathbf{coAM}$ from which our main result follows. Since L already has a constant-round proof system by assumption[4], $L \in \mathbf{AM}$. Thus, it suffices to show that $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}} \subseteq \mathbf{coAM}$. We do this by constructing a $\mathbf{AM}$ protocol for $\overline{L}$ where the strategy is to have the $\mathbf{AM}$ prover and verifier jointly simulate π and $\mathcal{V}^*_{\mathsf{GK}}$. In more detail, the $\mathbf{AM}$ verifier will pick the permutation π at random from a space of $\mathrm{poly}(T^m)$ permutations, where T is an upper bound on the running time of the reduction in the zero-knowledge protocol and m is the round complexity of the protocol; this turns out to suffice as a one-way permutation for the result in [21].[5] Next, we will have the $\mathbf{AM}$ prover and verifier jointly simulate each oracle computation of $\mathcal{V}^*_{\mathsf{GK}}$ using a (constant-round public-coin) random sampling protocol from [22]. Note that naively having the $\mathbf{AM}$ prover perform the computation of $\mathcal{V}^*_{\mathsf{GK}}$ fails for two reasons: a cheating $\mathbf{AM}$ prover may resample messages from a distribution different from the uniform distribution, and may not answer all of the $\mathcal{V}^*_{\mathsf{GK}}$ queries "independently". Finally, we rely on the constant adaptivity requirement of $(\mathcal{P}, \mathcal{V})$ to partially parallelize the executions of the random sampling protocol, so that the final protocol for $\overline{L}$ has constant round complexity.

As mentioned previously, in a recent work, Pass et al. [31] independently obtained results similar to ours. They also show that any language L for which there exists a fully black-box construction of constant-round zero-knowledge proofs from one-way functions is in $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$. Their techniques for doing this are dif-

[4] We can instantiate the protocol $(\mathcal{P}, \mathcal{V})$ for L with the identity permutation for this purpose.

[5] Having the $\mathbf{AM}$ verifier sample a random permutation "on the fly" does not work because the permutation π needs to be defined everywhere for $\mathcal{V}^*_{\mathsf{GK}}$ to be well-defined.

ferent from ours. They use a generic transformation from private-coin protocols into $\mathcal{V}^*_{\mathsf{GK}}$-relativized public-coin protocols, upon which the result then follows from the (relativized) lower bound for constant-round public-coin protocols in [15]. They then argue that if such proofs exist for all of **NP**, this would imply unlikely properties for the complexity class $\mathbf{BPP}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}$. Our techniques, on the other hand, allow us to relate the existence of such proofs to old questions in complexity such as whether $\mathbf{NP} \subseteq \mathbf{coAM}$ or whether **coNP** has interactive proofs with a $\mathbf{BPP}^{\mathbf{NP}}$ prover, whereas $\mathbf{BPP}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}$ is a new and less well-understood notion.

2 Preliminaries

2.1 Definitions

We let $[m] = \{1, \ldots, m\}$. For a random variable X, we let $x \leftarrow_{\mathrm{R}} X$ denote that x is sampled according to X. For a set S, we let $x \leftarrow_{\mathrm{R}} U_S$ denote x sampled according to the uniform distribution over S. We say that an event occurs with negligible probability if it occurs with probabilty $n^{-\omega(1)}$, and it occurs with overwhelming probability if it occurs with probability $1 - n^{-\omega(1)}$, where n is the input length.

Definition 1. *A permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is T-hard if for any circuit C of size at most T, and for y chosen uniformly at random,* $\Pr[C(y) = \pi^{-1}(y)] \leq \frac{1}{T}$*, where the probability is taken over the choice of y. If, given x, $\pi(x)$ is also efficiently computable then we call such a permutation a* one way permutation *(OWP).*

Definition 2. *Let Π_n be the set of all permutations from $\{0,1\}^n \to \{0,1\}^n$. Then, using the notation of [12], we define $\Pi_{k,n} \subseteq \Pi_n$ as $\{\pi_{k,n} \mid \pi_{k,n}(a,b) = (\pi_k(a), b)$ for some $\pi_k \in \Pi_k\}$ In other words, a uniform element of $\Pi_{k,n}$ is a random permutation on the first k bits, and fixes the last $n-k$ bits.*

Complexity Classes. We let $\mathbf{AM}[k]$ denote the class of languages that have $O(k)$-round public-coin interactive protocols (recall that public-coins are equivalent to private coins by [19]). Namely:

Definition 3. $L \in \mathbf{AM}[k]$ *if there is a $O(k)$-round public-coin interactive proof between an efficient verifier $\mathcal{V}$ and an all-powerful prover $\mathcal{P}$ such that:*

- *For all $x \in L$, $\mathcal{V}$ always accepts when interacting with $\mathcal{P}$.*
- *For all $x \notin L$ and all possibly cheating prover strategies $\mathcal{P}^*$, $\mathcal{V}$ accepts when interacting with $\mathcal{P}^*$ with only negligible probability.*

We let $\mathbf{AM} = \mathbf{AM}[O(1)]$. We say that a protocol $(\mathcal{P}, \mathcal{V})$ has an *honest prover strategy* of complexity $\mathcal{C}$ if the prover algorithm can be implemented by a machine in the class $\mathcal{C}$. We recall that **coNP** is in $\mathbf{AM}[n]$ with an honest prover strategy complexity of $\mathbf{P}^{\#\mathbf{P}}$ [27], and it is an open question whether the round complexity or the honest prover strategy complexity can be improved.

For any oracle $\mathcal{O}$, we let $\mathbf{BPP}^{\mathcal{O}[k]}$ denote the class of languages that are decidable by efficient randomized algorithms using at most k rounds of adaptive queries to an oracle $\mathcal{O}$. One round of adaptivity is a set of queries $x_1, \ldots, x_k$ the algorithm asks to the oracle such that the x_i can only depend on oracle answers to queries asked in previous rounds.

2.2 Zero-knowledge

In what follows we define a fully black-box construction of weak computational zero knowledge (CZK) from one way permutations. For a more general definition of CZK we refer the reader to previous literature [13]. As usual, we let $\mathsf{negl}(n)$ be some function such that $\mathsf{negl}(n) < \frac{1}{p(n)}$ for all polynomials $p(n)$.

Notation: we will use the following notation for interactive protocols. For any interactive protocol between a prover P and a verifier V, we let $2m$ denote the total number of rounds of communication, where a round consists of one message, either from P to V or from V to P. We let α_i denote the i^{th} message sent from P to V, and β_i the i^{th} response from V to P. Note that α_i is sent in round $2i-1$ and β_i is sent in round $2i$. Also, having P always send the first message is without loss of generality as we can set $\alpha_1 = \perp$ to model a proof where V goes first. For $i \in \{1 \ldots, m\}$, we let $\alpha_{[i]} = (\alpha_1, \ldots, \alpha_i)$. Let $V = (V_1, \ldots V_m)$ be the decomposition of V into its next-message functions. Here $V_i(x, \alpha_{[i]}, \omega)$ outputs β_i, the ith message sent by V when using input x, random coins ω, and receiving messages $\alpha_{[i]}$ from P. Let $\langle P, V\rangle(x)$ denote the verifier's view of an execution of the interactive protocol on an input x. This view includes all messages $\alpha_{[m]}$ sent by the prover, the verifier's random coins ω, and (if V is allowed access to an oracle) the answers to any oracle queries V may have made. We say that $\langle P, V\rangle(x)$ accepts if $V_m(x, \alpha_{[m]}, \omega) = 1$.

We will reserve calligraphic $\mathcal{P}, \mathcal{V}, \mathcal{S}$ to denote the prover, verifier, and simulator in a zero-knowledge protocol, and regular P, V to denote the prover and verifier in a (possibly non-zero-knowledge) interactive protocol.

Definition 4. *A fully black-box construction of a (weak) computational zero-knowledge proof system from one-way permutations for a language L is a tuple of oracle procedures $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ such that there exists a polynomial $T(n)$ satisfying the following properties for every family of permutations $\pi = \{\pi_n\}_{n \geq 1}$:*

Efficiency. *The running times of $\mathcal{V}, \mathcal{S}, M$ are bounded by $T = T(n)$.*

Completeness. *For all $x \in L$: $\Pr[\langle \mathcal{P}^\pi, \mathcal{V}^\pi\rangle(x) \text{ accepts}] \geq 1 - \mathsf{negl}(n)$.*

Soundness. *For all $x \notin L$ and for all (possibly computationally unbounded) $\mathcal{P}^*$,*

$$\Pr[\langle \mathcal{P}^*, \mathcal{V}^\pi\rangle(x) \text{ accepts}] \leq \mathsf{negl}(n).$$

Black-Box Zero-Knowledge. *For all (possibly computationally unbounded) $\mathcal{V}^*, D$ and for all $x \in L$: if*

$$\left|\Pr[D(\langle \mathcal{P}^\pi, \mathcal{V}^* \rangle(x)) = 1] - \Pr[D(\mathcal{S}^{\pi,\mathcal{V}^*}(x)) = 1]\right| > 1/n$$

then M can invert π, namely:

$$\Pr_{y \in \{0,1\}^n}[M^{\pi,\mathcal{V}^*,D}(y) = \pi^{-1}(y)] > 1/T$$

We note that completeness and soundness hold even if the given permutations are not one-way. Also, $\mathcal{V}^*, D$ are quantified after π is fixed and therefore may depend on π.

Comparison with standard definitions of zero-knowledge: The property that makes the above definition *weak* zero knowledge is that we only require the distinguishing advantage to be smaller than $1/n$, rather than negligible (the choice of $1/n$ was arbitrary; any non-negligible function will do). This enables us to consider simulators that run in *strict* polynomial time; it is known that in the standard definition of zero knowledge where the distinguishing advantage is negligible, no strict polynomial-time black-box simulators exist for constant-round protocols [3], although there are examples of non-black-box simulators [2]. It is useful for us to consider strict polynomial-time simulators because defining adaptivity is more straight-forward for such simulators than for expected polynomial-time simulators. This is discussed in the next section.

Nevertheless, we note here that any zero knowledge proof satisfying the standard definition also satisfies the weak definition above: if a simulator $\mathcal{S}'$ satisfies the standard definition and runs in expected time T', then a simulator $\mathcal{S}$ satisfies the weak definition by running $\mathcal{S}'$ for at most $2nT'$ steps, and halting with a failure symbol if $\mathcal{S}'$ does not produce an output in that time. By ruling out black-box constructions of weak zero knowledge proofs from one-way permutations, we also rule out proofs satisfying the standard definition. We note that the same discussion applies to the runtime of the reduction algorithm M.

Simplifying assumptions: we assume for simplicity that on inputs of length n, $\mathcal{V}$ and $\mathcal{S}$ only query π on inputs of length n. We assume that in an honest interaction of the protocol, the last message is from the verifier $\mathcal{V}$ to the prover $\mathcal{P}$ and contains the verifier's random coins. Clearly this does not affect either zero knowledge or soundness since it occurs after all "meaningful" messages are sent. This assumption allows us to define a transcript to be accepting if the honest verifier would accept that transcript using the coins output in the last message, and this definition remains meaningful even for transcripts generated by cheating verifiers. We assume without loss of generality that the simulator $\mathcal{S}$ never asks the same query twice and that it only asks *refinement* queries. Namely, for $i > 1$ and for every query $\alpha_{[i]} = (\alpha_{[i-1]}, \alpha_i)$ that the simulator queries to its cheating verifier black box $\mathcal{V}^*$, it must have previously queried $\alpha_{[i-1]}$ as well. We direct the reader to [14] for a discussion of why this holds without loss of generality.

2.3 Adaptivity

Here we define the *adaptivity* of the simulator, namely how much it uses responses from previous queries to the verifier black-box in order to generate new queries. All of the black-box simulators for constant-round zero knowledge in the literature intuitively work the following way: repeatedly query the cheating verifier with dummy queries enough times until it leaks some secret, then rewind and use this secret to output a simulated transcript [14, 5, 8, 9, 35]. The simulator may use the verifier's answers to determine whether to continue with dummy queries or to proceed to the next step of the simulation. If the simulator runs in *expected polynomial time* (rather than strict polynomial time), this procedure lasts indefinitely, making it hard to define the degree of the simulator's adaptivity. This is why we choose to work with *weak* zero knowledge, where the simulation is strict polynomial time; the definition of adaptivity becomes much simpler and more intuitive in this setting. We stress again that this only strengthens our result, as any zero-knowledge proof system satisfying the standard definition also satisfies the weak definition.

Definition 5. *A simulator $\mathcal{S}$ running in time T is said to be t-adaptive if it can be decomposed into $t+1$ oracle machines $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_t, \mathcal{S}_{t+1})$ with the following structure. Let x, ω (respectively) be the input and random coins for $\mathcal{S}$. For all permutations π and all cheating verifiers $\mathcal{V}^*$, $\mathcal{S}^{\pi,\mathcal{V}^*}$ operates as follows:*

1. *$\mathcal{S}_1^{\pi,\mathcal{V}^*}(x;\omega)$ generates at most T queries $q_1^{(1)}, \dots, q_T^{(1)}$ using x, ω. It sends these queries to $\mathcal{V}^*$ and gets back answers $\boldsymbol{a}_1 = (a_1^{(1)}, \dots, a_T^{(1)})$.*
2. *For each phase $j, 1 < j \leq t$, $\mathcal{S}_j^{\pi,\mathcal{V}^*}(x;\omega,\boldsymbol{a}_{j-1})$ generates at most T queries $q_1^{(j)}, \dots, q_T^{(j)}$ using x, ω and $\boldsymbol{a}_{j-1}$ which is the concatenation of all oracle answers from phases $1, \dots, j-1$. $\mathcal{S}_j^{\pi,\mathcal{V}^*}$ sets $\boldsymbol{a}_j$ to be the oracle answers $a_1^{(j)}, \dots, a_T^{(j)}$ for the j'th phase, concatenated with $\boldsymbol{a}_{j-1}$.*
3. *After obtaining $\boldsymbol{a}_t$, $\mathcal{S}_{t+1}^{\pi}(x;\omega,\boldsymbol{a}_t)$ computes the final output (notice it does so without calling $\mathcal{V}^*$).*

2.4 The Sam Oracle

Here we provide a description of the Sam oracle as defined in [21]. A more formal description can be found in [21].

Description of Sam_d: Sam_d takes as input a query $q = (i, C_{\mathsf{next}}, C_{\mathsf{prev}}, z)$ and outputs (ω', z'), such that:

1. ω' is chosen uniformly at random from:
   - the domain of C_{next} if $i = 1$.
   - the set $\{\omega \mid C_{\mathsf{prev}}(\omega) = z\}$ if $i > 1$.
2. $z' = C_{\mathsf{next}}(\omega')$.

The inputs to Sam_d are subject to the following restrictions:

1. The root query in every tree must include a security parameter 1^n such that $d = d(n)$ is the maximum depth query.
2. Queries with $i > d$ receive output $\perp$.
3. If $i > 1$, then the input $(i-1, C_{\mathsf{prev}}, \cdot, \cdot)$ was previously queried and resulted in output (ω, z) for some ω. Note that this restriction imposes a forest structure on the queries.
4. C_{next} is a *refinement* of C_{prev}. Formally: $C_{\mathsf{next}} = (C_{\mathsf{prev}}, \widetilde{C})$ for some circuit $\widetilde{C}$.

For our purposes, it is easier to think of Sam_d as being stateful, in which case the above restrictions can easily be implemented. Technically however Sam_d must be stateless, and so the above restrictions are enforced in [21] by giving Sam_d access to a signature protocol, and having him sign the output to every query, as well as the depth of the query, before returning a response. New queries are required to include a signature on a prior query, demonstrating that the first and third requirements have been met. (The refinement property can be verified by Sam_d independently.) Any query not meeting these restrictions receives output $\perp$. We direct the reader to [21] for the complete details (see also [22] for a precise statement about how to remove state), and we work with a stateful Sam_d for the remainder of this paper.

We will also consider Sam_d in a relativized world with a random permutation $\pi = \{\pi_n\}_{n\in\mathbb{N}}$, where $\pi_n : \{0,1\}^n \to \{0,1\}^n$ is chosen at random from all permutations mapping $\{0,1\}^n \to \{0,1\}^n$. We let Sam_d^π denote Sam_d in this relativized world. Sam_d^π is defined exactly as Sam_d, except it accepts circuits $C^\pi_{\mathsf{prev}}, C^\pi_{\mathsf{next}}$ that can possibly contain π gates.

We will abuse notation and write Sam to denote Sam_d for some $d = O(1)$. Our results will apply to all constant d so this slight abuse does not affect the correctness of our statements.

Using a Prover to Simulate Sam. Let $\mathbf{BPP}^{\mathsf{Sam}[t]}$ denote the class of languages that can be decided efficiently by a machine making at most t adaptive rounds of queries to the oracle Sam. We use the following theorem from [22] which shows that one can simulate this Sam oracle by a constant-round public-coin protocol.

Theorem 1 ([22]). *For any $L \in \mathbf{BPP}^{\mathsf{Sam}[t]}$, it holds that both L and $\overline{L}$ have $\mathbf{AM}[t]$ proofs with an honest prover strategy complexity of $\mathbf{BPP}^{\mathbf{NP}}$.*

3 Proof of Main Theorem

3.1 Overview

As discussed in the Introduction, our proof involves using a particular cheating verifier, $\mathcal{V}^*_{\mathsf{GK}}$ defined in Section 3.2, with the following properties:

- $\mathcal{V}^*_{\mathsf{GK}}$ cannot invert a random permutation π. This implies that the view $\langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathsf{GK}} \rangle(x)$ can be simulated by a simulator $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}(x)$ whenever $x \in L$. (Section 3.3)
- The simulator $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}(x)$ cannot produce an accepting transcript whenever $x \notin L$. Together with the previous property, this gives a way of deciding L. (Section 3.3)
- One can efficiently generate a transcript according to $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}(x)$ in a constant number of rounds with the help of an all-powerful (but possibly cheating) prover. Since, using the output of $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}(x)$, one can efficiently decide whether or not $x \in L$, this implies $L \in \mathbf{AM} \cap \mathbf{coAM}$. (Section 3.4)

3.2 Defining $\mathcal{V}^*_{\mathsf{GK}}$

Informally, upon receiving a message, the cheating verifier uniformly chooses a new random tape consistent with the transcript seen so far, and uses this to compute his next message. The formal definition follows, using notation defined in Section 2.1.

Fix any black-box construction of weak zero knowledge from one-way permutations $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$. Let $\omega \in \{0,1\}^T$ be a random tape for the honest verifier $\mathcal{V}$ which is divided into next-message functions $\mathcal{V}_1, \ldots, \mathcal{V}_m$. Define

$$R^{\alpha_{[i]}}_{\omega} = \{\omega' \in \{0,1\}^T \mid \forall j < i,\ \mathcal{V}_j(x, \alpha_{[j]}; \omega) = \mathcal{V}_j(x, \alpha_{[j]}; \omega')\} \tag{3.1}$$

i.e. the set of random tapes that, given prover messages $\alpha_{[i]}$, produce the same verifier messages as the random tape ω. For the special case where $i = 1$, set $R^{\alpha_1}_{\omega} = \{0,1\}^T$ for all α_1 and all ω.

Define $\mathcal{V}_{[i]} = (\mathcal{V}_1, \ldots, \mathcal{V}_i)$ to be the circuit that outputs the concatenation of $\mathcal{V}_1, \ldots, \mathcal{V}_i$. Namely, for every $\alpha_{[i]}$ and ω, it holds that

$$\mathcal{V}_{[i]}(\alpha_{[i]}, \omega) = (\mathcal{V}_1(\alpha_1, \omega), \mathcal{V}_2(\alpha_{[2]}, \omega), \ldots, \mathcal{V}_i(\alpha_{[i]}, \omega))$$

For any $\alpha_{[i]}$, let $\mathcal{V}_{[i]}(\alpha_{[i]}, \cdot)$ denote the circuit $\mathcal{V}_{[i]}$ with the input $\alpha_{[i]}$ hard-wired (therefore it takes only input ω.

Definition 6. *The cheating verifier $\mathcal{V}^*_{\mathsf{GK}} = (\mathcal{V}^*_{\mathsf{GK},1}, \ldots \mathcal{V}^*_{\mathsf{GK},m})$ is defined using the Sam^π_m oracle and a look-up table that associates server queries $\alpha_{[i]}$ with Sam^π_m oracle responses (ω, z). We write $\mathcal{V}^*_{\mathsf{GK}}$ with the understanding that the input x is hardwired into the verifier and the verifier is allowed oracle access to the permutation π and Sam^π_m.*

- $\mathcal{V}^*_{\mathsf{GK},1}(\alpha_1)$*: invoke $\mathsf{Sam}^\pi_m(1, \mathcal{V}_1(\alpha_1, \cdot), 0, 0)$ and let (ω_1, β_1) be the response. (Here, the 0 inputs are placeholders and can be replaced by anything.) Store $(\alpha_1, \omega_1, \beta_1)$ in the look-up table and output β_1.*
- $\mathcal{V}^*_{\mathsf{GK},i}(\alpha_{[i]})$ *for $i > 1$: let $\alpha_{[i]} = (\alpha_{[i-1]}, \alpha_i)$. Look up the value $(\alpha_{[i-1]}, \omega_{i-1}, \beta_{[i-1]})$ stored during a previous query. Query*

$$\mathsf{Sam}^\pi_m(i, \quad \mathcal{V}_{[i]}(\alpha_{[i]}, \ \cdot\ ), \quad \mathcal{V}_{[i-1]}(\alpha_{[i-1]}, \ \cdot\ ), \quad \beta_{[i-1]})$$

and let $(\omega_i, \beta_{[i]})$ be the response. Store $(\alpha_{[i]}, \omega_i, \beta_{[i]})$ in the look-up table and output β_i.

Observe that querying Sam_m^π in the manner described above for the case $i > 1$ returns an ω_i that is distributed uniformly in $R_{\omega_{i-1}}^{\alpha[i]}$.

Recall that we assume the simulator never repeats queries and only makes refinement queries. Therefore, $\mathcal{V}_{\mathsf{GK}}^*$ never tries to store inconsistent entries in the table, and $\mathcal{V}_{\mathsf{GK}}^*$ never queries its table for entries that do not exist. Therefore, $\mathcal{V}_{\mathsf{GK}}^*$'s queries to Sam_m^π always satisfy the restrictions laid out in Section 2.4. Observe that the output of $\langle \mathcal{P}^\pi, \mathcal{V}_{\mathsf{GK}}^* \rangle(x)$ is distributed identically to the honest $\langle \mathcal{P}^\pi, \mathcal{V}^\pi \rangle(x)$.

To complete the description of $\mathcal{V}_{\mathsf{GK}}^*$ we also need to construct a one-way permutation that remains one-way in the presence of a $\mathcal{V}_{\mathsf{GK}}^*$-oracle. To accomplish this, we refer to a result of Haitner et al. [21], which ruled out fully black-box constructions of $o(n/\log n)$-round statistically hiding commitment schemes form one-way permutations (where n is the security parameter). Building on and generalizing the works of [12, 36, 37], they demonstrated that by choosing π from $\Pi_{k,n}$ for appropriate k, π remains one-way even in the presence of a Sam_m^π-oracle.

Formally, the following lemma follows directly from their results.

Lemma 1 (implicit in [21]). *Suppose T, k satisfy $T^{3m+2} < 2^{k/8}$. Then, for any oracle machine A running in time T, it holds that:*

$$\Pr_{\pi \leftarrow_R \Pi_{k,n}, y \leftarrow_R U_n} [A^{\pi, \mathcal{V}_{\mathsf{GK}}^*}(y) = \pi^{-1}(y)] \leq 1/T$$

Proof. This follows from [21, Theorem 5.1], which established the above statement where $\mathcal{V}_{\mathsf{GK}}^*$ is replaced by Sam_m^π. From our definition of $\mathcal{V}_{\mathsf{GK}}^*$, it is clear that one call to $\mathcal{V}_{\mathsf{GK}}^*$ can be implemented using one call to Sam_m^π Furthermore, as noted above, since we assume $\mathcal{S}$ only makes unique refinement queries, all of the queries that $\mathcal{V}_{\mathsf{GK}}^*$ asks of Sam_m^π satisfy the restrictions in the definition of Sam_m^π.

3.3 Deciding L Using $\mathcal{V}_{\mathsf{GK}}^*$

We prove that $\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}(x)$ generates an accepting transcript with high probability if and only if $x \in L$.

Lemma 2. *Given any fully black-box construction from one-way functions of a constant-round weak zero knowledge proof $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ for a language L, and any n, k satisfying $T^{3m+2} < 2^{k/16}$, where $2m = O(1)$ is the round complexity of the proof system and $T = poly(n)$ is the strict polynomial bound on the running times of $\mathcal{V}, \mathcal{S}, M$, the following hold:*

1. *If $x \in L$, then $\Pr_{\pi \leftarrow_R \Pi_{k,n}, \mathcal{S}, \mathcal{V}_{\mathsf{GK}}^*}[\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$ generates accepting transcript$] \geq 2/3$.*
2. *If $x \notin L$, then $\Pr_{\pi \leftarrow_R \Pi_{k,n}, \mathcal{S}, \mathcal{V}_{\mathsf{GK}}^*}[\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$ generates accepting transcript$] \leq 1/3$.*

Proof.
Yes instances: We use the zero-knowledge property of the proof system to prove that for all $x \in L$:

$$\Pr[\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}(x) \text{ outputs an accepting transcript}] \geq 2/3 \tag{3.2}$$

The proof proceeds by contradiction, showing that if $\mathcal{S}$ fails to output an accepting transcript with sufficiently high probability then, by the weak zero-knowledge property of $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$, M can invert a random permutation $\pi \in \Pi_{k,n}$.

As was noted before, the distributions $\langle \mathcal{P}^{\pi}, \mathcal{V}^*_{\mathsf{GK}} \rangle(x) = \langle \mathcal{P}^{\pi}, \mathcal{V}^{\pi} \rangle(x)$. Therefore, by the completeness of the proof system, for $x \in L$, the transcript $\langle \mathcal{P}^{\pi}, \mathcal{V}^*_{\mathsf{GK}} \rangle(x)$ is accepted by the honest verifier with probability $1 - \mathsf{negl}(n)$. More formally, $\Pr[\mathcal{V}^{\pi}_m(x, \langle \mathcal{P}^{\pi}, \mathcal{V}^*_{\mathsf{GK}} \rangle(x)) = 1] \geq 1 - \mathsf{negl}(n)$.

For the sake of contradiction, assume that $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ outputs an accepting transcript with probability less than 2/3. That is, $\Pr[\mathcal{V}^{\pi}_m(x, \mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)) = 1] < 2/3$. Then we can use the honest verifier $\mathcal{V}$ to distinguish between the prover and simulator output, since $|\Pr[\mathcal{V}^{\pi}_m(x, \langle \mathcal{P}^{\pi}, \mathcal{V}^*_{\mathsf{GK}} \rangle) = 1] - \Pr[\mathcal{V}^{\pi}_m(x, \mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)) = 1]| > 1/3 - \mathsf{negl}(n)$. Therefore, by the weak black-box zero-knowledge property of $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$, there exists an oracle machine $M^{\pi, \mathcal{V}^*_{\mathsf{GK}}, \mathcal{V}}$ running in time T that can break the one-wayness of π with probability at least $1/T$. We can remove oracle access to $\mathcal{V}$ by having M simulate $\mathcal{V}$ by making at most T oracle calls to π for each call to $\mathcal{V}$. Thus, we get a machine $M^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$ running in time T^2 such that $\Pr_{\pi \leftarrow_{\mathrm{R}} \Pi_{k,n}, y \leftarrow_{\mathrm{R}} U_n}[M^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(y) = \pi^{-1}(y)] \geq 1/T > 1/T^2$. This yields a contradiction to Lemma 1, and Equation (3.2) follows.

No instances: here, we use statistical soundness (following [23, 26, 15]) to argue that for all $x \notin L$:

$$\Pr[\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x) \text{ outputs an accepting transcript}] \leq 1/3 \qquad (3.3)$$

The proof proceeds by contradiction, showing that if $\mathcal{S}$ outputs an accepting transcript with high probability, then there exists a cheating prover $\mathcal{P}^*_{\mathsf{GK}}$ that breaks the statistical soundness of the proof system. Let T, the running time of $\mathcal{S}$, be the bound on the total number of $\mathcal{V}^*_{\mathsf{GK}}$ queries made by $\mathcal{S}$, and let m be the round complexity of the zero knowledge proof system. Starting from $\mathcal{V}^*_{\mathsf{GK}}$, we define a new (inefficient) prover strategy $\mathcal{P}^*_{\mathsf{GK}}$ which interacts with an external verifier $\mathcal{V}$ as follows:

1. Choose queries to forward to $\mathcal{V}$: On input x, $\mathcal{P}^*_{\mathsf{GK}}$ picks a random subset of query indices $U = \{j_1, j_2, \ldots, j_m\} \subset [T]$ of size m. The set U represents the queries that $\mathcal{P}^*_{\mathsf{GK}}$ will forward to the verifier $\mathcal{V}$.
2. Simulate $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$: Internally simulate $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ step by step. We handle the j'th oracle query, q_j, that $\mathcal{S}$ makes to $\mathcal{V}^*_{\mathsf{GK}}$ as follows. Let $q_j = \alpha_{[i]}$ for some $i \leq m$.
   – If $j \notin U$: Simulate $\mathcal{V}^*_{\mathsf{GK}}$ internally to answer q_j. More formally, look up the value $(\alpha_{[i-1]}, \omega)$ stored during a previous $\mathcal{V}^*_{\mathsf{GK}}$ query. (Note that since $\mathcal{S}$ only makes refinement queries, $\mathcal{S}$ must have made such a query.) Choose $\omega' \leftarrow R^{\alpha_{[i]}}_{\omega}$ uniformly at random ($\mathcal{P}^*_{\mathsf{GK}}$ can do this since he is computationally unbounded), store $(\alpha_{[i]}, \omega')$ and output $\mathcal{V}_i(x, \alpha_{[i]}, \omega')$.
   – If $j \in U$: If $q_j = \alpha_{[i]}$ and $i > 1$, forward α_i to the external $\mathcal{V}$. Upon receiving β_i in response, look up the stored value $(\alpha_{[i-1]}, \omega)$ and uniformly sample a random string $\omega'' \leftarrow \{\omega' \in R^{\alpha_{[i]}}_{\omega} \wedge \mathcal{V}_i(x, \alpha_{[i]}, \omega') = \beta_i\}$. Store $(\alpha_{[i]}, \omega'')$ and output β_i.

Note that as long as $\mathcal{S}$ outputs an accepting transcript with noticeable probability when interacting with $\mathcal{V}^*_{\mathsf{GK}}$ on $x \notin L$ then this cheating prover $\mathcal{P}^*_{\mathsf{GK}}$ has a noticeable probability of outputting an accepting transcript when interacting with the honest verifier $\mathcal{V}$. This happens if $\mathcal{P}^*_{\mathsf{GK}}$ chooses U to include exactly the messages that are used by $\mathcal{S}$ in his output. $\mathcal{P}^*_{\mathsf{GK}}$ succeeds in choosing the correct queries with probability at least $1/T^{O(m)}$. Thus, if $\mathcal{S}$ outputs an accepting transcript with probability $> 1/3$ then $\mathcal{P}^*_{\mathsf{GK}}$ outputs an accepting transcript with probability at least $1/3 \cdot 1/T^{O(m)}$ which is non-negligible when $m = O(1)$. This is a contradiction of the fact that the proof has negligible soundness error, thus (3.3) follows.

3.4 Applying Theorem 1 To Remove $\mathcal{V}^*_{\mathsf{GK}}$

We can now combine Lemma 2 and Theorem 1 to prove our main theorem.

Theorem 2. *Suppose there is a black-box construction from a one-way permutation of a constant-round weak zero knowledge proof* $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ *for a language* L*, where* $\mathcal{S}$ *is* t*-adaptive. Then both* L *and* $\overline{L}$ *are in* $\mathbf{AM}[t]$ *with honest prover complexity* $\mathbf{BPP}^{\mathbf{NP}}$*.*

Proof. From Lemma 2 we already know that $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$ decides L. We will construct an oracle algorithm A based on $\mathcal{S}$, such that A^{Sam} decides L and furthermore the adaptivity of A is the same as the adaptivity of $\mathcal{S}$.

Sampling π *Efficiently:* By Lemma 1, we know that for π to be one-way in the presence of $\mathcal{V}^*_{\mathsf{GK}}$, it is sufficient to choose $\pi \leftarrow_{\mathrm{R}} \Pi_{k,n}$ with $k = 9(3m+2)\log T = O(\log n)$. Such a permutation can be sampled in polynomial time by sampling a uniform permutation on $k = O(\log n)$ bits. Let $A_1^{\mathcal{V}^*_{\mathsf{GK}}}$ be identical to $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$, except A_1 first samples π itself and then runs $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$.

From Definition 6, it holds that oracle access to $\mathcal{V}^*_{\mathsf{GK}}$ can be implemented using oracle access to Sam and an additional look-up table to associate previous queries with previous oracle responses. Therefore there exists a reduction A^{Sam} that on all inputs behaves identically to $A_1^{\mathcal{V}^*_{\mathsf{GK}}}$, and furthermore the adaptivity of A is identical to the adaptivity of A_1, whose adaptivity in turn is the same as that of $\mathcal{S}$.

Since $\mathcal{S}$ has adaptivity t, this implies that $L \in \mathbf{BPP}^{\mathsf{Sam}[t]}$. We can therefore apply Theorem 1 to conclude the proof.

4 Acknowledgements

We would like to thank Jonathan Katz for helpful discussions.

References

1. Aiello, W., Hastad, J.: Statistical zero-knowledge languages can be recognized in two rounds. JCSS 42, 327–345 (1991)

2. Barak, B.: How to go beyond the black-box simulation barrier. In: Proc. 42nd FOCS. pp. 106–115. IEEE (2001)
3. Barak, B., Lindell, Y.: Strict polynomial-time in simulation and extraction. In: STOC. pp. 484–493 (2002)
4. Barak, B., Lindell, Y., Vadhan, S.: Lower bounds for non-black-box zero knowledge. JCSS 72(2), 321–391 (2006)
5. Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: EUROCRYPT. pp. 280–305 (1997)
6. Bellare, M., Micali, S., Ostrovsky, R.: Perfect zero-knowledge in constant rounds. In: STOC. pp. 482–493 (1990)
7. Blum, M.: How to prove a theorem so no one else can claim it. In: Proc. ICM (1986)
8. Brassard, G., Crépeau, C., Yung, M.: Everything in NP can be argued in *perfect* zero-knowledge in a *bounded* number of rounds. In: Eurocrypt '89. pp. 192–195 (1989), lNCS No. 434
9. Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Crypto '89. pp. 526–545 (1989), lNCS No. 435
10. Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: CRYPTO. pp. 526–544 (1989)
11. Fortnow, L.: The complexity of perfect zero-knowledge. In: STOC '87. pp. 204–209 (1987)
12. Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM J. Comput. 35(1), 217–246 (2005)
13. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, New York, NY, USA (2004)
14. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptology 9(3), 167–190 (1996)
15. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. 25(1), 169–192 (1996)
16. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991), prelim. version in FOCS '86
17. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems 7(1), 1–32 (Winter 1994), preliminary version in FOCS' 87
18. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
19. Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. Advances in Computing Research: Randomness and Computation 5, 73–90 (1989)
20. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: CRYPTO. pp. 408–423 (1998)
21. Haitner, I., Hoch, J.J., Reingold, O., Segev, G.: Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In: Proc. FOCS '07. pp. 669–679 (2007)
22. Haitner, I., Mahmoody-Ghidary, M., Xiao, D.: A new sampling protocol and applications to basing cryptography on **NP**-hardnss. In: Proc. CCC 2010 (2010), to appear. Full version available as ECCC TR-867-09
23. Haitner, I., Reingold, O., Vadhan, S., Wee, H.: Inaccessible entropy. In: STOC. pp. 611–620 (2009)

24. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. of Com. 28(4), 1364–1396 (1999), preliminary versions appeared in STOC' 89 and STOC' 90.
25. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: STOC. pp. 44–61 (1989)
26. Katz, J.: Which languages have 4-round zero-knowledge proofs? In: TCC. pp. 73–88 (2008)
27. Lund, C., Fortnow, L., Karloff, H.J., Nisan, N.: Algebraic methods for interactive proof systems. In: FOCS. pp. 2–10 (1990)
28. Naor, M.: Bit commitment using pseudorandomness 4(2), 151–158 (1991), preliminary version in CRYPTO' 89
29. Ostrovsky, R., Wigderson, A.: One-way functions are essential for non-trivial zero-knowledge. In: ISTCS '93. pp. 3–17 (1993)
30. Pass, R.: Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on np-hardness. In: IEEE Conference on Computational Complexity. pp. 96–110 (2006)
31. Pass, R., Venkitasubramaniam, M.: Private coins versus public coins in zero-knowledge proof systems. In: TCC. pp. 588–605 (2010)
32. Pass, R., Wee, H.: Black-box constructions of two-party protocols from one-way functions. In: TCC. pp. 403–418 (2009)
33. Pavan, A., Selman, A.L., Sengupta, S., Vinodchandran, N.V.: Polylogarithmic-round interactive proofs for conp collapse the exponential hierarchy. Theor. Comput. Sci. 385(1-3), 167–178 (2007)
34. Reingold, O., Trevisan, L., Vadhan, S.: Notions of reducibility between cryptographic primitives. In: Proc. 1st TCC. pp. 1–20 (2004)
35. Rosen, A.: A note on constant-round zero-knowledge proofs for np. In: Naor, M. (ed.) TCC. Lecture Notes in Computer Science, vol. 2951, pp. 191–202. Springer (2004)
36. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Proc. EUROCRYPT '98. vol. 1403, pp. 334–345 (1998)
37. Wee, H.: One-way permutations, interactive hashing and statistically hiding commitments. In: TCC. pp. 419–433 (2007)