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

S. Dov Gordon[*] Hoeteck Wee[†] David Xiao[‡] Arkady Yerukhimovich[*]

Abstract

We consider the following problem: can we construct constant-round zero-knowledge proofs (with negligible soundness) for $\mathbf{NP}$ 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 $\mathbf{coAM}$ have constant-round zero-knowledge proofs of this kind.

[*] Dept. of Computer Science, University of Maryland.

[†] Queens College, CUNY. Most of this research was completed while a post-doc at Columbia University.

[‡] LRI, Université Paris-Sud. This reseach was completed while the author was 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 [20], 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 $\mathbf{NP}$ (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 $\mathbf{NP}$ under the minimal assumption of one-way functions [10, 31]. 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 [17, 7], and the minimal assumption from which we know how to construct constant-round computational zero-knowledge proofs for $\mathbf{NP}$ is constant-round statistically *hiding* commitments [15], which seem to be a stronger assumption than one-way permutations [39, 23]. 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 [30, 26]. This raises the following intriguing open problem:

> Can we base constant-round zero-knowledge proofs for $\mathbf{NP}$ 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, 22], 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 $\mathbf{BPP}$ such as graph isomorphism and graph non-isomorphism [17, 6], all languages with statistical zero knowledge proofs lie in $\mathbf{AM} \cap \mathbf{coAM}$ [1, 12] (and therefore do not include all of $\mathbf{NP}$ unless the polynomial hierarchy collapses).

Lower bounds. Goldreich and Krawczyk [16] showed that 3-round zero-knowledge protocols and public-coin constant-round zero-knowledge protocols with black-box simulators exist only for languages in $\mathbf{BPP}$. Katz [28] showed that 4-round zero-knowledge proofs only exist for languages in $\mathbf{MA} \cap \mathbf{coMA}$. Haitner et al. [23] 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 [15]. More recently, Haitner et al. [25] established a partial converse to [15], namely that any constant-round zero-knowledge proof for $\mathbf{NP}$ 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 $\mathbf{NP}$ 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 $\mathbf{NP}$:

> **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. [36, 27]) 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 $\mathbf{NP}$.

- 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 $\mathbf{NP}$. [11, 34].

- We have unconditional constructions of constant-round statistical black-box zero-knowledge proofs for graph isomorphism and graph non-isomorphism, languages which are in $\mathbf{AM} \cap \mathbf{coAM}$ but are commonly believed to lie outside $\mathbf{BPP}$.

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 [18, 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: the formal statement of our result (Theorem 3.3) is slightly more general as we can obtain non-trivial consequences even when the simulator's adaptivity is super-constant but sufficiently smaller than linear. We defer discussion of these strengthenings to Section 3.4.

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 $\mathbf{AM} \cap \mathbf{coAM}$. The high level strategy is to extend the Goldreich-Krawczyk lower bound for constant-round public-coin protocols [16] to the private-coin setting. Following [16] (also [32, 28, 25]), we consider a cheating verifier $\mathcal{V}^*_{\mathrm{GK}}$ 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 $\mathcal{V}^*_{\mathrm{GK}}$ efficiently for general private-coin protocols. The computational complexity of $\mathcal{V}^*_{\mathrm{GK}}$ comes up in two ways in [16]: first to deduce that the zero-knowledge property holds against such a $\mathcal{V}^*_{\mathrm{GK}}$, 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. [23], which, roughly speaking, demonstrates the existence of a one-way permutation π secure in the presence of a $\mathcal{V}^*_{\mathrm{GK}}$ 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 $\mathcal{V}^*_{\mathrm{GK}}$ since π is one-way against $\mathcal{V}^*_{\mathrm{GK}}$. Following [16, 28, 25], we may then deduce a $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}$ algorithm for L. (Such a statement was obtained independently by Pass and Venkitasubramaniam [33].[1]) Along the way, we will exploit (as with [28, 25]) 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 $\mathcal{V}^*_{\mathrm{GK}}$.

Next, we will essentially show that $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathrm{GK}}} \subseteq \mathbf{AM} \cap \mathbf{coAM}$ from which our main result follows. Since L already has a constant-round proof system by assumption[2], $L \in \mathbf{AM}$. Thus, it suffices to show that $\mathbf{BPP}^{\pi, \mathcal{V}^*_{\mathrm{GK}}} \subseteq \mathbf{coAM}$. 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 $\mathcal{V}^*_{\mathrm{GK}}$. 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 [23].[3] Next, we will have the $\mathbf{AM}$ prover and verifier jointly simulate each oracle computation of $\mathcal{V}^*_{\mathrm{GK}}$ using a (constant-round public-coin) random sampling protocol from [24]. Note that naively having the $\mathbf{AM}$ prover perform the computation of $\mathcal{V}^*_{\mathrm{GK}}$ 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 $\mathcal{V}^*_{\mathrm{GK}}$ 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.

2 Preliminaries

2.1 Definitions

Definition 2.1. 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 2.2. Let Π_n be the set of all permutations from $\{0,1\}^n \to \{0,1\}^n$. Then, using the notation of [13], 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

[1] They obtained the result via a generic transformation from private-coin protocols into Sam-relativized public-coin protocols, upon which the result then follows from the (relativized) lower bound for constant-round public-coin protocols in [16].

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

[3] 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 $\mathcal{V}^*_{\mathrm{GK}}$ to be well-defined.

of $\Pi_{k,n}$ is a random permutation on the first k bits, and fixes the last $n-k$ bits.

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 [14]. 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 2.3. 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 [15] 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 [15, 5, 9, 10]. 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 2.4. A simulator $\mathcal{S}$ running in time T is said to be t-adaptive if it can be decomposed into $t+1$ oracle machines $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_t, \mathcal{S}_{t+1})$ 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 $q_1^{(1)}, \ldots, q_T^{(1)}$ using x, ω. It sends these queries to $\mathcal{V}^*$ and gets back answers $\vec{a}_1 = (a_1^{(1)}, \ldots, a_T^{(1)})$.

2. For each phase $j, 1 < j \leq t$, $\mathcal{S}_j^{\pi,\mathcal{V}^*}(x;\omega,\vec{a}_{j-1})$ generates at most T queries $q_1^{(j)}, \ldots, q_T^{(j)}$ using x, ω and $\vec{a}_{j-1}$ which is the concatenation of all oracle answers from phases $1, \ldots, j-1$. $\mathcal{S}_j^{\pi,\mathcal{V}^*}$ sets $\vec{a}_j$ to be the oracle answers $a_1^{(j)}, \ldots, a_T^{(j)}$ for the j'th phase, concatenated with $\vec{a}_{j-1}$.

3. After obtaining $\vec{a}_t$, $\mathcal{S}_{t+1}^{\pi}(x;\omega,\vec{a}_t)$ computes the final output (notice it does so without calling $\mathcal{V}^*$).

2.4 The $\mathsf{SampleWithSize}$ protocol

In our proof, we will require a constant-round public-coin sampling protocol from [24]. The verifier desires to sample a random element from an efficiently decidable set R using the help of an all-powerful prover, assuming that the verifier knows (an approximation of) $|R|$. Such protocols have a long history in theoretical computer science [38, 21, 12, 1], but for us the most relevant prior work is Goldreich *et al.* [19].

The $\mathsf{SampleWithSize}$ protocol of [24] also guarantees that if the verifier knows s that is a multiplicative approximation of $|R|$, then the verifier can obtain a sample x distributed close to uniform in R. In addition, given a partition of R, the $\mathsf{SampleWithSize}$ provides a *size* s_x that approximates the size of the set in the partition that x belongs to. We will use this protocol in Section 3.4 recursively: the size parameter s_x is used with $\mathsf{SampleWithSize}$ to sample from the partition that x belongs to, and so forth.

Theorem 2.5 (Theorem 4.1 of [24])**.** *There exists a constant-round public-coin protocol* $\mathsf{SampleWithSize} = (P_{\mathsf{SWS}}, V_{\mathsf{SWS}})$ *whose parties get as input:*

1. *A set $R \subset \{0,1\}^n$ represented as a polynomial-size circuit deciding membership in R.*

2. *A positive integer $s \in \mathbb{N}$, satisfying $s \in [(1 \pm (\frac{\delta}{100n})^8)|R|]$*

3. *A partition Λ of R given by a labeling function $\Lambda : R \to \{0,1\}^*$ such that two elements $x, x' \in R$ are in the same partition set iff $\Lambda(x) = \Lambda(x')$,* i.e. *they share the same label. Λ is given as a polynomial-size circuit.*

4. *A security parameter $\delta \geq 1/\mathrm{poly}(n)$ given in unary.*

Let $\langle P^*, V_{\mathsf{SWS}}\rangle(R, s, \Lambda, \delta)$ denote the interaction of the verifier with any prover P^*, resulting either in an output (x, s_x) or "abort". The protocol achieves the following guarantee:

1. ***Completeness:*** *an interaction with the honest prover $\langle P_{\mathsf{SWS}}, V_{\mathsf{SWS}}\rangle(R, s, \Lambda, \delta)$ aborts with probability at most δ.*

2. ***Correctness and soundness:*** *when interacting with any (honest or deviating) prover P^*, either (conditioned on not aborting) the output $(x, s_x) = \langle P^*, V_{\mathsf{SWS}}\rangle(R, s, \Lambda, \delta)$ satisfies both the following:*

   (a) *With probability at least $1 - \delta$, $s_x \in [(1 \pm \delta)|\Lambda^{-1}(\Lambda(x))|]$.*

   (b) *Conditioned on $s_x \in [(1 \pm \delta)|\Lambda^{-1}(\Lambda(x))|]$, x is distributed δ-close to uniform over R.*

   or else $\langle P^*, V_{\mathsf{SWS}}\rangle(R, s, \Lambda, \delta)$ aborts with probability at least $1 - \delta$.

$\mathsf{SampleWithSize}$ is applied in [24] in a way that is similar to the setting of this paper, *i.e.* it is used to replace oracle calls to a sampling oracle by invocations of this protocol.

3 Proof of main theorem

3.1 Overview

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

- $\mathcal{V}^*_{\mathrm{GK}}$ cannot invert a random permutation π. This implies that the view $\langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathrm{GK}}\rangle(x)$ can be simulated by a simulator $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}(x)$ whenever $x \in L$. (Section 3.3)

- The simulator $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$ 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 $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$ in a constant number of rounds with the help of an all-powerful (but possibly cheating) prover. Since one can efficiently decide using the output of $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$ whether or not $x \in L$, this implies $L \in \mathbf{AM} \cap \mathbf{coAM}$. (Section 3.4)

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

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 $\mathcal{V}_1, \dots, \mathcal{V}_m$, then 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. The cheating verifier $\mathcal{V}^*_{\mathrm{GK}} = (\mathcal{V}^*_{\mathrm{GK},1}, \dots \mathcal{V}^*_{\mathrm{GK},m})$ is defined below. To simplify notation, we omit the input x (which is fixed), the permutation π (which is sampled once and then fixed), and the random coins of $\mathcal{V}^*_{\mathrm{GK}}$.

- $\mathcal{V}^*_{\mathrm{GK},1}(\alpha_1)$: choose a random tape ω uniformly at random, store (α_1, ω) and output $\mathcal{V}_1(\alpha_1, \omega)$.

- $\mathcal{V}^*_{\mathrm{GK},i}(\alpha_{[i]})$ for $i > 1$: let $\alpha_{[i]} = (\alpha_{[i-1]}, \alpha_i)$. Look up the value $(\alpha_{[i-1]}, \omega)$ stored during a previous query. Choose $\omega' \leftarrow R^{\alpha_{[i]}}_{\omega}$ uniformly at random, store $(\alpha_{[i]}, \omega')$ and output $\mathcal{V}_i(\alpha_{[i]}, \omega')$.

Recall that we assume the simulator never repeats queries and only makes refinement queries. Therefore, $\mathcal{V}^*_{\mathrm{GK}}$ never tries to store inconsistent entries in the table, and $\mathcal{V}^*_{\mathrm{GK}}$ never queries its table for entries that do not exist. Observe that the output of $\langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathrm{GK}} \rangle(x)$ is distributed identically to the honest $\langle \mathcal{P}^\pi, \mathcal{V}^\pi \rangle(x)$. However, $\mathcal{V}^*_{\mathrm{GK}}$ is not necessarily efficient, since there may be no way to sample from $R^{\alpha_{[i]}}_{\omega}$ efficiently.

We want to construct a one-way permutation that remains one-way in the presence of a $\mathcal{V}^*_{\mathrm{GK}}$-oracle. To accomplish this, we refer to a result of Haitner et al. [23], which ruled out fully black-box constructions of $\Omega(n/\log n)$-round statistically hiding commitment schemes form one-way permutations (where n is the security parameter). More generally and somewhat informally, they (building on and generalizing the works of [13, 37, 39]) demonstrated oracles π, Sam with the following properties:

- π is a random permutation on k bits and is one-way in the presence of a Sam-oracle, and

- Sam can be used to instantiate a cheating sender that breaks the binding property of any m-round statistically hiding commitment scheme, as long as $k \gtrsim m \log T$, where T is the running time of the security reduction.

Moreover, the afore-mentioned cheating sender uses a "resampling" strategy exactly as $\mathcal{V}^*_{\mathrm{GK}}$ does, and therefore, the Sam oracle can be used to efficiently instantiate our cheating verifier strategy. Haitner et al. prove that a sufficently large random permutation π remains one-way in the presence of Sam. The following lemma follows directly from their results.

Lemma 3.1 (implicit in [23]). *Suppose T, k satisfy $T^{3m+2} < 2^{k/8}$. Then, for any oracle machine R running in time T, it holds that:*

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

Proof. This is essentially a special case of [23, Theorem 5.1], which established the above statement where $\mathcal{V}^*_{\mathrm{GK}}$ is replaced with a so-called sampling oracle Sam (refer to Appendix A or to [23] for the formal specification of Sam). It is straight-forward to verify that $\mathcal{V}^*_{\mathrm{GK}}$ is a special instance of the Sam oracle; while Sam resamples preimages in arbitrary circuits, $\mathcal{V}^*_{\mathrm{GK}}$ only resamples preimages in the circuits computing the honest verifier's next message function.[4] Furthermore, the restrictions that Sam imposes on its queries are imposed by $\mathcal{V}^*_{\mathrm{GK}}$ as well. In particular, we assume $\mathcal{S}$ only makes refinement queries, and since there are only m prover messages in a protocol, the longest sequence of refinement queries that $\mathcal{S}$ may ask is of length m. In the notation of [23], then, we only allow queries of $depth(n) = m$. ■

3.3 Deciding L using $\mathcal{V}^*_{\mathbf{GK}}$

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

Lemma 3.2. *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 $\Pr_{\pi \leftarrow_R \Pi_{k,n}, \mathcal{S}, \mathcal{V}^*_{\mathrm{GK}}}[\mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}$ generates accepting transcript$] \geq 2/3$.*
2. *If $x \notin L$, then $\Pr_{\pi \leftarrow_R \Pi_{k,n}, \mathcal{S}, \mathcal{V}^*_{\mathrm{GK}}}[\mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}$ generates accepting transcript$] \leq 1/3$.*

Proof. We only prove the case of yes instances; no instances are treated exactly as in the argument of [16] and we defer it to the appendix.

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

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

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

For the sake of contradiction, assume that $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}(x)$ outputs an accepting transcript with probability less than $2/3$. That is, $\Pr[\mathcal{V}^\pi_m(x, \mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}(x)) = 1] < 2/3$. Then we can use the honest verifier $\mathcal{V}$ to distinguish between the prover and simulator output, since $|\Pr[\mathcal{V}^\pi_m(x, \langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathrm{GK}} \rangle) = 1] - \Pr[\mathcal{V}^\pi_m(x, \mathcal{S}^{\pi, \mathcal{V}^*_{\mathrm{GK}}}(x)) = 1]| > 1/3 - \mathsf{negl}(n)$. Therefore, by the weak black-box zero-knowledge property of $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$, there exists an oracle machine $M^{\pi, \mathcal{V}^*_{\mathrm{GK}}, \mathcal{V}}$ running in time T that can break the one-wayness of π with probability at least $1/T$. We

[4] More specifically, letting $\mathcal{V}^\pi_{x,\alpha[i]}(\omega) = \mathcal{V}^\pi_i(x, \alpha_i, \omega)$, where the first two inputs are hardwired into the circuit, a valid Sam query is (roughly) of the form $(\mathcal{V}^\pi_{x,\alpha[i]}, \mathcal{V}^\pi_{x,\alpha[i-1]}, \beta_{i-1})$, where some prior query to Sam, $(\mathcal{V}^\pi_{x,\alpha[i-1]}, \mathcal{V}^\pi_{x,\alpha[i-2]}, \beta_{i-2})$, resulted in output (ω, β_{i-1}) such that $\mathcal{V}^\pi_{x,\alpha[i-1]}(\omega) = \beta_{i-1}$. Furthermore, because we assume that all simulator queries are refinement queries, the query $(\mathcal{V}^\pi_{x,\alpha[i]}, \mathcal{V}^\pi_{x,\alpha'[i-1]}, \beta_{i-1})$ can only occur if $\alpha_{[i]} = (\alpha'_{[i-1]}, \alpha'_i)$, as required by Sam.

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 $M^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ running in time T^2 such that $\Pr_{\pi\in\Pi_{k,n},y\leftarrow_{\mathrm{R}}U_n}[M^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(y) = \pi^{-1}(y)] \geq 1/T > 1/T^2$. This yields a contradiction to Lemma 3.1, and (3.2) follows.

■

3.4 Removing π and $\mathcal{V}^*_{\mathbf{GK}}$

Theorem 3.3. *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 $L \in \mathbf{AM} \cap \mathbf{coAM}[t]$.*

Proof. We construct a $O(t)$-round interactive proof $(P_{\mathcal{S}}, V_{\mathcal{S}})$ for the complement language $\overline{L}$, which proves that $L \in \mathbf{coAM}[t]$. The high-level idea is for $V_{\mathcal{S}}$ to emulate the computation of $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ and to use the all-powerful prover to answer any queries the simulator makes to $\mathcal{V}^*_{\mathrm{GK}}$. If the prover were always honest it would answer these queries as $\mathcal{V}^*_{\mathrm{GK}}$ would, allowing $V_{\mathcal{S}}$ to emulate $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ exactly, and enabling him to decide L. (Recall from Lemma 3.2 that with high probability $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$ produces an accepting transcript iff $x \in L$.)

Two problems arise here. The first is that the prover may deviate from the prescribed behavior. To handle this, we replace each query to $\mathcal{V}^*_{\mathrm{GK}}$ with an execution of the $\mathsf{SampleWithSize}$ sampling protocol from Theorem 2.5. Recall that this protocol catches a cheating prover with high probability. The second problem is that $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ makes $O(T)$ queries to $\mathcal{V}^*_{\mathrm{GK}}$; if we ran these $\mathsf{SampleWithSize}$ protocols sequentially, we would have a $O(T)$-round protocol for L. Instead, we will exploit the t-adaptivity of $\mathcal{S}$ to parallelize the executions of $\mathsf{SampleWithSize}$. Recall from Definition 2.4 that $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ can be decomposed into $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}} = (\mathcal{S}_1^{\pi,\mathcal{V}^*_{\mathrm{GK}}}, \ldots, \mathcal{S}^{\pi}_{t+1})$, each of which makes only parallel queries to $\mathcal{V}^*_{\mathrm{GK}}$. To attain an $O(t)$-round protocol for $\overline{L}$, for each adaptive step $j \in [t]$, we will execute in parallel all of the $\mathsf{SampleWithSize}$ executions corresponding to the queries made by $\mathcal{S}_j$.

Sampling π efficiently: our first observation is that since $\pi \leftarrow_{\mathrm{R}} \Pi_{k,n}$ and $k = 9(3m+2)\log T = O(\log n)$, such a permutation can be sampled efficiently by $V_{\mathcal{S}}$. Therefore, in the first step of the proof, our verifier $V_{\mathcal{S}}$ will sample such a π and send it to the prover, and they then both use this fixed π for the rest of the proof.

The AM protocol for $\overline{L}$: The main tool we use is the protocol $\mathsf{SampleWithSize}$ from [24], which was given in Theorem 2.5. We construct an $\mathbf{AM}$ proof system $(P_{\mathcal{S}}, V_{\mathcal{S}})$ for $\overline{L}$ that uses this protocol to emulate the computation of $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$.

First we set up some notation. Let $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}} = (\mathcal{S}_1^{\pi,\mathcal{V}^*_{\mathrm{GK}}}, \ldots, \mathcal{S}_t^{\pi,\mathcal{V}^*_{\mathrm{GK}}}, \mathcal{S}^{\pi}_{t+1})$ denote the decomposition of the t-adaptive simulator as described in Definition 2.4. Since $\mathcal{S}$'s running time is bounded by T, for all $j \in [t]$, $\mathcal{S}_j$ makes at most T oracle queries. Since for the rest of this proof we will be working with a fixed x and a permutation π chosen once and then fixed, we will let $\mathcal{V}$ denote the honest verifier with these fixed choices of x, π, and let $\mathcal{V} = (\mathcal{V}_1, \ldots, \mathcal{V}_m)$ its decomposition into next-message functions. Assume without loss of generality that the simulator's adaptivity t is at least m, which is half the number of rounds of the zero knowledge proof and the number of messages sent by $\mathcal{V}$.

Define the error loss function $\mathsf{loss}(\delta) = (\frac{\delta}{100n})^8$, and let $\mathsf{loss}^0(\delta) = \delta$ and $\mathsf{loss}^i(\delta) = \mathsf{loss}^{i-1}(\mathsf{loss}(\delta))$. Then for $i \in [t]$, we define the error parameter $\delta_i = \mathsf{loss}^{m-i}(1/(ntT))$. It is easy to check that $\delta_{i-1} = (\frac{\delta_i}{100n})^8$ for all i and furthermore $\delta_1 = (1/ntT)^{O(8^m)} = 1/\mathrm{poly}(n)$ as long as $m = O(1)$.

We will use the definition of $R_\omega^{\alpha[i]}$ given in Equation 3.1.

Protocol 3.4. The following protocol $(P_{\mathcal{S}}, V_{\mathcal{S}})$ takes input x of length n, and the verifier $V_{\mathcal{S}}$ either accepts or rejects by performing the following:

1. $V_{\mathcal{S}}$ samples $\pi \leftarrow_{\mathrm{R}} \Pi_{k,n}$ and sends a description of π to $P_{\mathcal{S}}$.

2. $V_{\mathcal{S}}$ allocates a lookup table Table that associates $\mathcal{S}$'s queries $\alpha_{[i]}$ with entries (ω, s). The table is initially empty.

3. $V_{\mathcal{S}}$ samples random coins for $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ and uses these random coins to emulate $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ as follows. Sequentially for each adaptive step $j = 1$ through $j = t$, $V_{\mathcal{S}}$ emulates the computation of $\mathcal{S}_j^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ to compute queries $\alpha^{(j)}_{[i_1]}, \ldots, \alpha^{(j)}_{[i_T]}$. $V_{\mathcal{S}}$ engages in the following sub-protocol for all the queries in step j in parallel. Let $\alpha_{[i]}$ be one such query:

   (a) If $i = 1$ then $V_{\mathcal{S}}$ sets $\omega_0 = 0^n, s_0 = 2^T$. If $i > 1$, then $V_{\mathcal{S}}$ looks up $\mathsf{Table}(\alpha_{[i-1]}) = (\omega_{i-1}, s_{i-1})$.

   (b) $V_{\mathcal{S}}$ sets $\Lambda_{\alpha_{[i]}}$ to be the circuit taking input ω and outputting $\mathcal{V}_i(\alpha_{[i]}; \omega)$.

   (c) $V_{\mathcal{S}}$ engages in the $\mathsf{SampleWithSize}$ protocol with the prover on input $R^{\alpha_{[i]}}_{\omega_{i-1}}, s_{i-1}, \Lambda_{\alpha_{[i]}}, \delta_i$.

   (d) If $\mathsf{SampleWithSize}$ aborts then $V_{\mathcal{S}}$ aborts, otherwise $V_{\mathcal{S}}$ attains an output (ω_i, s_i), records $\mathsf{Table}(\alpha_{[i]}) = (\omega_i, s_i)$ and returns $\beta_{[i]} = \mathcal{V}_i(\alpha_{[i]}; \omega_i)$.

4. After t adaptive steps, $V_{\mathcal{S}}$ runs $\mathcal{S}^{\pi}_{t+1}$ on the obtained query responses in order to output a verifier view τ.

5. Using the honest verifier algorithm $\mathcal{V}$, $V_{\mathcal{S}}$ checks where τ is an accepting view. If it is rejecting then $V_{\mathcal{S}}$ accepts (since we want to decide the complement of L), otherwise $V_{\mathcal{S}}$ rejects.

$V_{\mathcal{S}}$'s efficiency is clear. It is also clear that the protocol runs in $O(t)$ rounds. Completeness and soundness follow from the following claim:

Claim 3.5. *If $V_{\mathcal{S}}$ emulates $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$ using* $\mathsf{SampleWithSize}$ *on input x as described above, then both of the following hold:*

1. ***Completeness:** The probability $\langle P_{\mathcal{S}}, V_{\mathcal{S}} \rangle$ aborts is at most $1/n$.*

2. ***Correctness and Soundness:** For any (honest or cheating) prover P^*, let τ denote the distribution of views generated by $V_{\mathcal{S}}$ in Step 4 of Protocol 3.4 when $V_{\mathcal{S}}$ interacts with P^*. Then, either (conditioned on $V_{\mathcal{S}}$ not aborting) the statistical distance satisfies*

$$\Delta(\tau, \mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)) \leq 2/n$$

   or else $\langle P^*, V_{\mathcal{S}} \rangle(x)$ aborts with probability $1 - 1/n$.

First we use Claim 3.5 to prove that Protocol 3.4 is a complete and sound $\mathbf{AM}$ protocol for $\overline{L}$, then we turn to proving the claim.

Completeness of Protocol 3.4: If $x \in \overline{L}$, then by the first item of Claim 3.5 the honest prover causes $V_{\mathcal{S}}$ to abort with probability at most $1/n$, and by the second item, conditioned on not aborting, τ is $2/n$-statistically close to $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$. Since $x \in \overline{L}$, Lemma 3.2 says that $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$ produces a rejecting transcript with probability at least $2/3$. It follows that τ is a rejecting transcript with probability at least $2/3 - 2/n$. Since $V_{\mathcal{S}}$ accepts when τ is a rejecting transcript, $V_{\mathcal{S}}$ will accept with probability at least $2/3 - 3/n$.

Soundness of Protocol 3.4: If $x \notin \overline{L}$, then by the second item of Claim 3.5, any prover either causes $V_{\mathcal{S}}$ to abort with probability $1 - 1/n$, or, conditioned on not aborting, it holds that τ is $2/n$-statistically close to $\mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)$. As in the completeness case, combining this with Lemma 3.2 implies that $V_{\mathcal{S}}$ accepts with probability at most $1/3 + 2/n$. Regardless of the prover's strategy, the verifier accepts with probability at most $1/3 + 2/n$. ■

3.4.1 Proof of Claim 3.5

Completeness The only time $V_{\mathcal{S}}$ might abort is in an invocation of $\mathsf{SampleWithSize}$. Recall that an execution with an honest prover ends in abort with probability at most $\delta_i \leq \delta_m$. Since there are at most tT invocations of $\mathsf{SampleWithSize}$, the probability that the honest prover causes any invocation to abort is at most $\delta_m tT \leq 1/n$.

Soundness We will use the definition of the set $R_\omega^{\alpha[i]}$ as given in Equation 3.1. Let us say that the good event $Good_j$ occurs if for all $j \in [t]$ the following is true for all invocations of $\mathsf{SampleWithSize}$ in $V_{\mathcal{S}}$'s emulation of $\mathcal{S}_1^{\pi,\mathcal{V}^*_{\mathrm{GK}}}, \ldots, \mathcal{S}_j^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$:

- $V_{\mathcal{S}}$ has not aborted.
- Let $\alpha_{[i]}$ be any query that was made in adaptive step j', where $1 \leq j' \leq j$. Then the response (ω, s_ω) obtained by running $\mathsf{SampleWithSize}$ with the prover satisfies $s_\omega \in [(1 \pm \delta_i)|R_\omega^{\alpha[i]}|]$.
- Conditioned on the first two items, ω is δ_i-close to uniformly distributed in $R_\omega^{\alpha[i]}$.

Let us define $Good_0$ to hold always. Notice that if $Good_j$ holds, then $Good_{j'}$ holds for all $j' < j$.

Also define $Aborts_j$ to be the event that $V_{\mathcal{S}}$ aborts somewhere in the emulation of $\mathcal{S}_1^{\pi,\mathcal{V}^*_{\mathrm{GK}}}, \ldots, \mathcal{S}_j^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$. Notice that $\overline{Aborts_t}$ is the event that $V_{\mathcal{S}}$ never aborts in the entire protocol.

Claim 3.6. *Fix a prover strategy P^*. Either for all $j \in [t]$ it holds that*

$$\Pr[Good_j \mid Good_{j-1} \wedge \overline{Aborts_j}] \geq 1 - 1/(nt) \tag{3.3}$$

or else $\Pr[Aborts_t] \geq 1 - 1/n$.

Using Claim 3.6 to prove Claim 3.5: Suppose $\Pr[Aborts_t] = \Pr[V_{\mathcal{S}} \text{ aborts}] < 1 - 1/n$ (since otherwise we are done). Then, since $Good_0$ always holds, Claim 3.6 followed by a union bound over all steps implies that

$$\Pr[Good_t \mid \overline{Aborts_t}] \geq 1 - 1/n \tag{3.4}$$

Since $\delta_i \leq \delta_m$, it holds for all $j \leq t$ that conditioned on event $Good_j$, the outputs of the invocations of $\mathsf{SampleWithSize}$ are δ_m-statistically close to the outputs of $\mathcal{V}^*_{\mathrm{GK}}$. Since there are at most tT queries, by the triangle inequality their joint distribution is at most $tT\delta_m = 1/n$ statistically far from the output of $\mathcal{V}^*_{\mathrm{GK}}$. The input x, the simulator's random coins, and the answers to the queries completely determine the output of the simulator. Therefore

$$\Delta((\tau \mid Good_t), \mathcal{S}^{\pi,\mathcal{V}^*_{\mathrm{GK}}}(x)) \leq 1/n \tag{3.5}$$

Combining Inequality 3.4 and Inequality 3.5, we obtain Claim 3.5.

Proving Claim 3.6: Let P^* be an arbitrary, possibly cheating prover strategy. Notice that this prover strategy may use dependencies between all the parallel instances of $\mathsf{SampleWithSize}$ that are invoked in step j. However, the soundness of $\mathsf{SampleWithSize}$ holds against all (possibly unbounded) prover strategies, so soundness still holds in each of the parallel instances of $\mathsf{SampleWithSize}$. Therefore we can analyze each instance separately.

We prove Claim 3.6 by induction on the steps j. Inequality 3.3 holds for $Good_0$, since we defined $Good_0$ to always be true, therefore the base case is trivial. For the inductive case, Suppose Inequality 3.3 holds for all $j' \leq j - 1$. In the j'th step for $j \geq 1$, $V_{\mathcal{S}}$ invokes the $\mathsf{SampleWithSize}$ protocol for all of $\mathcal{S}_j^{\pi,\mathcal{V}^*_{\mathrm{GK}}}$'s queries, conditioned on $Good_{j-1}$. This conditioning implies that for all previous invocations $\alpha_{[i]}$ that were answered with (ω, s_ω), it holds that $s_\omega \in [(1 \pm \delta_i)|R_\omega^{\alpha[i]}|]$. Let us say that such an output is δ_i-accurate.

The overall prover strategy P^* induces prover strategies for each of the T parallel invocations of $\mathsf{SampleWithSize}$ generated by $\mathcal{S}_j^{\pi, \mathcal{V}^*_{\mathrm{GK}}}$. Since we invoked $\mathsf{SampleWithSize}$ with error parameter δ_i, if at least one such invocation aborts with probability $\geq 1 - \delta_i > 1 - 1/n$, then the overall protocol aborts with at least this probability.

Suppose that $V_{\mathcal{S}}$ does not abort with such high probability. Let $\alpha_{[i]}$ be any query in the j'th adaptive step. Let $\alpha_{[i-1]}$ be the prefix of $\alpha_{[i]}$, then because $\mathcal{S}$ only asks refinement queries we know that $V_{\mathcal{S}}$ already queried $\alpha_{[i-1]}$ and furthermore because $Good_{j-1}$ holds it has recorded a δ_{i-1}-accurate answer (ω, s_ω) for $\alpha_{[i-1]}$ in its table. By definition, $\delta_{i-1} = (\frac{\delta_i}{100n})^8$, therefore using the fact that (ω, s_ω) is δ_{i-1}-accurate for $\alpha_{[i-1]}$, $V_{\mathcal{S}}$ invokes $\mathsf{SampleWithSize}$ with s_ω that is a good enough approximation of $R_\omega^{\alpha[i]}$. Therefore Theorem 2.5 implies the following two cases:

Case 1: there exists some query $\alpha_{[i]}$ made in the j'th adaptive step, such that $\mathsf{SampleWithSize}$ causes $V_{\mathcal{S}}$ to abort with probability $\geq 1 - \delta_i$. By the inductive hypothesis (Inequality 3.3 holds for all $1, \ldots, j-1$) and a union bound over all $1, \ldots, j-1$, it holds that

$$\Pr[Good_{j-1} \mid \overline{Aborts_{j-1}}] \geq 1 - (j-1)/(nt)$$

Therefore we can deduce that

$$\begin{aligned}
\Pr[V_{\mathcal{S}} \text{ aborts}] &\geq (1 - \Pr[Aborts_{j-1}]) \Pr\left[Aborts_j \mid \overline{Aborts_{j-1}}\right] + \Pr[Aborts_{j-1}] \\
&\geq \Pr\left[Aborts_j \wedge Good_{j-1} \mid \overline{Aborts_{j-1}}\right] \\
&\geq \Pr\left[Good_{j-1} \mid \overline{Aborts_{j-1}}\right] \Pr[Aborts_j \mid Good_{j-1}, \overline{Aborts_{j-1}}] \\
&\geq (1 - \tfrac{j-1}{nt})(1 - \delta_i) \\
&\geq 1 - 1/n
\end{aligned}$$

Case 2: for every query $\alpha_{[i]}$ in adaptive step j, the invocation of $\mathsf{SampleWithSize}$ on $\alpha_{[i]}$ causes $V_{\mathcal{S}}$ to abort with probability $< 1 - \delta_i$. In this case, by a union bound and the fact that $\delta_i \leq \delta_j$ for all queries in the j'th adaptive step, $V_{\mathcal{S}}$ aborts with probability $< 1 - \delta_j T$. Furthermore, the correctness condition of $\mathsf{SampleWithSize}$ implies that conditioned on not aborting both the following hold for every query $\alpha_{[i]}$ made in round j,

1. With probability $> 1 - \delta_i > 1 - \delta_j$, the output (ω, s_ω) of $\mathsf{SampleWithSize}$ is δ_i-accurate.
2. Conditioned on (ω, s_ω) being δ_i-accurate, it holds that ω is distributed δ_i-close to uniform in $R_\omega^{\alpha[i]}$

Therefore by a union bound over all T queries in round j, it holds that

$$\Pr[Good_j \mid Good_{j-1} \wedge \overline{Aborts_j}] \geq 1 - \delta_j T > 1 - 1/(nt)$$

■

Interpretation: Combining Theorem 3.3 with [8] it says that constant-round weak zero knowledge proofs for $\mathbf{NP}$ based on one-way permutations with a constant-adaptive simulator are unlikely to exist unless the polynomial hierarchy collapses; combining it with [35] it says that such proofs for $\mathbf{NP}$ with a $\mathrm{polylog}(n)$-adaptive simulator are unlikely to exist unless the *exponential* hierarchy collapses; and in general it says any such proof for $\mathbf{NP}$ with a $o(n)$-adaptive simulator would imply a new interactive protocol for $\mathbf{coNP}$ whose round complexity beats linear, which is the best known [29].

References

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

[2] B. Barak. How to go beyond the black-box simulation barrier. In *Proc.* 42*nd FOCS*, pages 106–115. IEEE, 2001.

[3] B. Barak and Y. Lindell. Strict polynomial-time in simulation and extraction. In *STOC*, pages 484–493, 2002.

[4] B. Barak, Y. Lindell, and S. Vadhan. Lower bounds for non-black-box zero knowledge. *JCSS*, 72(2):321–391, 2006.

[5] M. Bellare, M. Jakobsson, and M. Yung. Round-optimal zero-knowledge arguments based on any one-way function. In *EUROCRYPT*, pages 280–305, 1997.

[6] M. Bellare, S. Micali, and R. Ostrovsky. Perfect zero-knowledge in constant rounds. In *STOC*, pages 482–493, 1990.

[7] M. Blum. How to prove a theorem so no one else can claim it. In *Proc. ICM*, 1986.

[8] R. B. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[9] G. Brassard, C. Crépeau, and M. Yung. Everything in NP can be argued in *perfect* zero-knowledge in a *bounded* number of rounds. In *Eurocrypt '89*, pages 192–195, 1989. LNCS No. 434.

[10] U. Feige and A. Shamir. Zero knowledge proofs of knowledge in two rounds. In *Crypto '89*, pages 526–545, 1989. LNCS No. 435.

[11] U. Feige and A. Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO*, pages 526–544, 1989.

[12] L. Fortnow. The complexity of perfect zero-knowledge. In *STOC '87*, pages 204–209, 1987.

[13] R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

[14] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, New York, NY, USA, 2004.

[15] O. Goldreich and A. Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

[16] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[17] O. Goldreich, S. Micali, and A. Wigderson. 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.

[18] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. 7(1):1–32, Winter 1994. Preliminary version in FOCS' 87.

[19] O. Goldreich, S. Vadhan, and A. Wigderson. On interactive proofs with a laconic prover. In *Proc. 28th ICALP*, 2001.

[20] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[21] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989.

[22] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In *CRYPTO*, pages 408–423, 1998.

[23] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *Proc. FOCS '07*, pages 669–679, 2007.

[24] I. Haitner, M. Mahmoody-Ghidary, and D. Xiao. A constant-round public-coin protocol for sampling with size, and applications. Technical report, Princeton University, 2009. `http://www.cs.princeton.edu/~dxiao/docs/hash.pdf`.

[25] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *STOC*, pages 611–620, 2009.

[26] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. 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.

[27] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, 1989.

[28] J. Katz. Which languages have 4-round zero-knowledge proofs? In *TCC*, pages 73–88, 2008.

[29] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. pages 2–10, 1990.

[30] M. Naor. Bit commitment using pseudorandomness. 4(2):151–158, 1991. Preliminary version in CRYPTO' 89.

[31] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *ISTCS '93*, pages 3–17, 1993.

[32] R. Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on np-hardness. In *IEEE Conference on Computational Complexity*, pages 96–110, 2006.

[33] R. Pass and M. Venkitasubramaniam. Manuscript, 2009.

[34] R. Pass and H. Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.

[35] A. Pavan, A. L. Selman, S. Sengupta, and N. V. Vinodchandran. Polylogarithmic-round interactive proofs for conp collapse the exponential hierarchy. *Theor. Comput. Sci.*, 385(1-3):167–178, 2007.

[36] O. Reingold, L. Trevisan, and S. Vadhan. Notions of reducibility between cryptographic primitives. In *Proc. 1st TCC*, pages 1–20, 2004.

[37] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Proc. EUROCRYPT '98*, volume 1403, pages 334–345, 1998.

[38] L. G. Valiant and V. V. Vazirani. Np is as easy as detecting unique solutions. In *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 458–463, New York, NY, USA, 1985. ACM.

[39] H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *TCC*, pages 419–433, 2007.

A The Sam Oracle [23]

For completeness, we provide a description of the Sam oracle as defined in [23]. Sam is given access to a family of random permutations $\pi = \{\pi_n\}_{n=1}^{\infty}$ where $\pi_n \in \Pi_n$. Sam takes as input a circuit C^{π}_{prev} (possibly containing π-gates), and some output z of this circuit. It returns a uniform pre-image of z under C^{π}_{prev}. The challenge is to construct Sam such that Sam access enables uniform sampling of pre-images without enabling the user to invert π. To achieve this, several important restrictions on the queries are described below. A more formal description can be found in [23].

Description of Sam: Sam takes as input a query $q = (C^{\pi}_{\mathsf{next}}, C^{\pi}_{\mathsf{prev}}, z)$ and outputs (ω', z'), such that:

- ω' is chosen uniformly at random from the set $\{\omega \mid C^{\pi}_{\mathsf{prev}}(\omega) = z\}$.
- $z' = C^{\pi}_{\mathsf{next}}(\omega')$.

The restrictions are as follows:

1. The input $(C^{\pi}_{\mathsf{prev}}, \cdot, \cdot)$ was previously queried and resulted in output (ω, z). Note that this restriction imposes a forest structure on the queries.
2. C^{π}_{next} is a *refinement* of C^{π}_{prev}. Formally: $C^{\pi}_{\mathsf{next}}(\omega) = (C^{\pi}_{\mathsf{prev}}(\omega), \widetilde{C}^{\pi}(\omega))$ for some circuit $\widetilde{C}^{\pi}$ and for every ω.
3. The root query in every tree must include a security parameter 1^n such that for some function $\mathsf{depth} : \mathbb{N} \to \mathbb{N}$, $\mathsf{depth}(n)$ specifies the maximum depth of that particular tree. Queries in that tree can (only) have circuits containing π_j-gates for $j \leq n$, and queries beyond depth $\mathsf{depth}(n)$ receive output $\perp$.

Technically, the above restrictions are enforced in [23] by giving Sam 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 independently.) Any query not meeting these restrictions receives output $\perp$. We direct the reader to [23] for the complete details, and leave them implicit below.

We commented above that the first restriction imposes a forest structure on the queries. More specifically, a *root query* is of the form $(1^n, C^{\pi}_{\mathsf{next}}, \perp, \perp)$, and is answered by Sam with (ω', z'), where ω' chosen at uniformly at random from the domain of C^{π}_{next}, and $z' = C^{\pi}_{\mathsf{next}}(\omega')$. For all future legal queries of the form $(C^{\pi}_{\mathsf{next}}, C^{\pi}_{\mathsf{prev}}, z)$, the parent query is defined in the natural way: it is (the first) query of the form $(C^{\pi}_{\mathsf{prev}}, \cdot, \cdot)$ that resulted in output z. We say (informally) that a circuit queries Sam up to depth d if the maximum depth of any query tree is at most d.

B Proof of No instances

Here, we use statistical soundness (following [25, 28, 16]) to argue that for all $x \notin L$:

$$\Pr[\mathcal{S}^{\pi,\mathcal{V}^*_{\mathsf{GK}}}(x) \text{ outputs an accepting transcript}] \leq 1/3 \tag{2.1}$$

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 $\omega' \leftarrow R_\omega^{\alpha_{[i]}}$ 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 $\omega'' \leftarrow \{\omega' \in R_\omega^{\alpha_{[i]}} \wedge \mathcal{V}_i(x, \alpha_{[i]}, \omega') = \beta_i\}$. 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 (2.1) follows.