Which Languages Have 4-Round Zero-Knowledge Proofs?

Jonathan Katz[*]

Abstract

We show that if a language L has a 4-round, black-box, computational zero-knowledge proof system with negligible soundness error, then $\bar{L} \in \mathsf{MA}$. Assuming the polynomial hierarchy does not collapse, this means in particular that NP-complete languages do not have 4-round zero-knowledge proofs with black-box simulation.

1 Introduction

A zero-knowledge proof system [24] for a language L is a protocol that enables a prover $\mathcal{P}$ to convince a polynomial-time verifier $\mathcal{V}$ that a given instance x is indeed a member of L. Roughly speaking, the guarantees provided are:

Completeness: If $x \in L$ then the honest prover $\mathcal{P}$ will convince the honest verifier $\mathcal{V}$ to accept, except possibly with some small probability. If $\mathcal{P}$ always convinces $\mathcal{V}$ to accept when $x \in L$ then we say the proof system has *perfect* completeness.

Soundness: If $x \notin L$ a cheating prover $\mathcal{P}^*$ will be unable to falsely convince the honest verifier that x is in L, except with some small probability known as the *soundness error*.

Zero knowledge: When $x \in L$ and the prover is honest, even a malicious verifier $\mathcal{V}^*$ "learns nothing" beyond the fact that $x \in L$.

There are various ways of formalizing the above properties. In this paper, we do not require that the honest prover be implementable in polynomial time. We consider the case where soundness holds against all-powerful provers — i.e., we focus on *proofs* rather than *arguments* [14] — and are interested in proof systems with negligible soundness error. For a proof system to be non-trivial, the completeness error should not be too large; we will consider both the case of perfect completeness as well as the case when, for $x \in L$, the honest verifier accepts only with some noticeable (i.e., inverse polynomial) probability. Finally, we focus on the case of *computational* zero knowledge (CZK) where, informally, the requirement is that a non-uniform *polynomial-time* cheating verifier learns nothing from the interaction. (Formal definitions are provided in Section 2.) We let CZK denote the class of languages that admit a computational zero-knowledge proof system.

[*]Dept. of Computer Science, University of Maryland, jkatz@cs.umd.edu. This work was supported by NSF CAREER award #0447075 and US-Israel Binational Science Foundation grant #2004240.

In this paper we study the round complexity of CZK proof systems, where a round consists of a message sent from one party to the other and we assume that the prover and the verifier speak in alternating rounds. We survey what is known in this regard:

Unconditional constructions. The only languages currently known to be in CZK *unconditionally* are those that admit *statistical* zero-knowledge (SZK) proofs [24] where, informally, even an all-powerful cheating verifier learns nothing from its interaction with the prover; we denote the class of languages admitting statistical zero-knowledge proofs by SZK. It has recently been established [35] that all languages in SZK have constant-round statistical zero-knowledge proof systems (with negligible soundness error).[1] As particular (and chronologically earlier) special cases, graph non-isomorphism [22, Remark 12] as well as languages related to various number-theoretic problems [24, 32, 37, 16, 33, 15] have 4-round SZK proof systems, and graph isomorphism [8] has a 5-round SZK proof system.

Constructions based on one-way functions/permutations. Assuming the existence of one-way functions, every language in NP has an $\omega(1)$-round CZK proof system where the honest prover runs in polynomial time given an NP-witness for the statement being proved [22]. (Actually, this result holds for MA as well.[2]) If no computational restrictions are placed on the honest prover, then any language in AM has an $\omega(1)$-round CZK proof system under the same assumption, and any language in $\mathsf{IP} = \mathsf{PSPACE}$ has a CZK proof system with polynomially many rounds [31, 11].

Assuming the existence of one-way permutations, Feige and Shamir [18] show a 4-round computational zero-knowledge *argument* for any language in NP. Their techniques yield a 5-round CZK argument based on one-way functions; this was later improved to 4 rounds by Bellare et al. [7]. (A 4-round argument can also be constructed using the work of [6], based on one-way functions and mild complexity-theoretic assumptions.)

Constructions based on stronger assumptions. Assuming the existence of a two-round statistically hiding commitment scheme, there exists a 5-round CZK proof system for any language in NP [20], or even AM if the honest prover can be unbounded. (More generally, given a constant-round statistically hiding commitment scheme, there exists a constant-round CZK proof system for any language in AM.) Two-round statistically hiding commitment schemes, in turn, can be constructed based on a variety of number-theoretic assumptions [13, 14, 25] and, more generally, the existence of collision-resistant hash functions [17, 30].

Although statistically hiding commitment schemes can be constructed from any one-way function [29], black-box constructions of *constant-round* statistically hiding commitment schemes from one-way functions do not exist [28].

Lower bounds. Goldreich and Oren [23] show that 2-round CZK proofs exist only for languages in BPP. (Their result applies to *auxiliary-input* zero knowledge proofs, the type we will be concerned with here as well.) Extending this result, Goldreich and Krawczyk [21] show that 3-round *black-box* CZK proofs exist only for languages in BPP. (A definition of black-box CZK is given in Section 2.) Both these results hold for arguments as well as proofs.

[1] The constant-round proofs in [9] consider a weaker variant of SZK where the verifier is assumed to run in polynomial time during its interaction with the prover. See [38] for further discussion of these variants.

[2] MA is a randomized version of NP, and is defined in Section 2. AM denotes the class of languages having *constant-round* Arthur-Merlin proofs.

1.1 Our Result

We show that 4-round black-box CZK proofs, even with imperfect completeness, exist only for languages whose complement is in MA. This result is unconditional, and holds independent of any cryptographic assumptions one might make. Other than the fact that the bound holds only with respect to black-box simulation, this result is essentially the best one could hope for:

- Under widely believed number-theoretic assumptions, there do exist 5-round CZK proofs for all of NP [20]. Assuming the polynomial hierarchy does not collapse [12], our result indicates that the round complexity in this case is optimal.
- Our result applies only to proofs, but not arguments. Indeed, as noted earlier, there do exist 4-round CZK *arguments* for all of NP under relatively weak assumptions [18, 7].
- There exist 4-round SZK proofs for languages believed to be outside of BPP, such as graph non-isomorphism [22].

We remark also that for the case of *uniform* zero-knowledge (i.e., protocols which are zero knowledge for *uniform* polynomial-time verifiers), a 4-round protocol for all of NP is possible using the techniques of [20] and assuming the existence of 1-round statistically hiding commitment schemes (that are computationally binding for uniform adversaries). One advantage of considering non-uniform zero-knowledge (where the zero-knowledge property holds even for verifiers given arbitrary auxiliary input) is that such protocols remain zero knowledge under sequential composition [23].

Besides shedding further light on the finer structure of the class CZK, our result indicates that (black-box) 4-round CZK proofs for all of NP are impossible and so the round complexity of the Goldreich-Kahan protocol [20] is optimal. Our result also gives an "explanation" as to why the known SZK proof for graph isomorphism requires five rounds [8] while graph *non*-isomorphism has a 4-round SZK proof [22].

Limitations of black-box impossibility results. We prove our result only for the case of *black-box* zero-knowledge protocols (i.e., where the simulator is given only black-box access to the cheating verifier). The work of Barak [3], however, shows that black-box impossibility results and lower bounds need not carry over to the general case.[3] Nevertheless, black-box bounds are useful insofar as they rule out one class of approaches for solving a problem. We remark further that Barak's techniques currently yield only *arguments* rather than *proofs*; this is also the case for nonblack-box protocols based on "knowledge of exponent" assumptions [27, 10]. On the other hand, some nonblack-box zero-knowledge *proofs* using four or fewer rounds are known to exist based on non-standard assumptions [5, 34].

Our current ability to prove *general* (as opposed to black-box) lower bounds for zero-knowledge protocols is, unfortunately, relatively limited [23, 5].

1.2 A High-Level Overview of Our Technique

Our lower bound for 4-round protocols is proved by extending the Goldreich-Krawczyk lower bound [21] for 3-round protocols. (We assume familiarity with their proof in what follows.) To prove their result, Goldreich and Krawczyk consider a cheating verifier $\mathcal{V}^*$ who generates its message, in

[3] Barak's work gives a constant-round, public-coin, CZK argument for all of NP, something that was ruled out with respect to black-box simulation by Goldreich and Krawczyk [21].

the second round of the protocol, using fresh random coins that are determined as a function of the prover's first message. On an intuitive level this means that rewinding is useless because every time $\mathcal{V}^*$ is rewound, and a different first message is sent by the simulator, it is as if the protocol execution is being started again from scratch.

We use the same basic idea, now applied to the verifier's message sent in the *third* round of the protocol. A problem is that the verifier's first-round message may "commit" the verifier, in a computational sense, to only one possible third-round message. (Roughly speaking, the verifier cannot be committed in an information-theoretic sense because then an all-powerful prover could guess the third-round message *in advance* based on the first-round message alone. This is one reason why our result applies only to proofs, and not arguments.) For this reason, we need some "all-powerful" entity to provide the verifier with *collisions*, i.e., multiple third-round messages consistent with the same first-round message. This idea was inspired by the work of Haitner et al. [28], who use collisions of exactly this sort to prove lower bounds on the round complexity of black-box constructions of interactive protocols in other settings. In their work, an oracle provides collisions. Here, we do not have an oracle; instead, we have an all-powerful prover ("Merlin") provide such collisions as part of an interactive MA-proof for some language. See Section 3 for further intuition, as well as the details of the proof.

An easy extension of our results shows that if a language L has a 4-round CZK proof system where the verifier's first message is *independent* of the instance (and depends only on the instance *length*), then $\bar{L} \in \mathsf{P/poly}$. This explains why the 4-round SZK proof system for graph non-isomorphism [22] uses an instance-dependent message in the first round.

1.3 Outline of the Paper

Standard definitions, as well as some terminology specific to this paper, are provided in Section 2. In Section 3 we prove our result for the case of CZK proof systems with perfect completeness. Technical modifications necessary to deal with the case of imperfect completeness are deferred to Section 4. We conclude with some open questions in Section 5.

2 Definitions

Given interactive algorithms $\mathcal{P}$ and $\mathcal{V}$, we let $\langle \mathcal{P}(x), \mathcal{V}(y) \rangle$ denote the interaction of $\mathcal{P}$, holding input x, with $\mathcal{V}$, holding input y. We let $\langle \mathcal{P}(x), \mathcal{V}(y) \rangle = 1$ denote the event that $\mathcal{V}$ outputs 1 in the indicated interaction, where an output of "1" is interpreted as "accept" and an output of "0" is interpreted as "reject". We now give the standard definition of an interactive proof system [24] for a language L.

Definition 1 Interactive algorithms $\mathcal{P}, \mathcal{V}$ form an interactive proof system for a language L if $\mathcal{V}$ runs in probabilistic polynomial time and there exist functions c, s such that:

- For all $x \in L$, it holds that $\Pr[\langle \mathcal{P}(x), \mathcal{V}(x) \rangle = 1] \geq c(|x|)$.
- For all $x \notin L$ and any $\mathcal{P}^*$ we have $\Pr[\langle \mathcal{P}^*, \mathcal{V}(x) \rangle = 1] \leq s(|x|)$.
- There exists a polynomial p such that $c(|x|) \geq s(|x|) + 1/p(|x|)$.

(Note that we do not require $\mathcal{P}$ to run in polynomial time.) We call c the acceptance probability, and s the soundness error. If $c(|x|) = 1$ for all x, we say the proof system has perfect completeness. If s is negligible, we say the proof system has negligible soundness error. $\diamondsuit$

We will only consider zero-knowledge proof systems having negligible soundness error.

A *round* of an interactive proof system consists of a message sent from one party to the other, and we assume the prover and verifier speak in alternating rounds. Following [2], we let MA denote the class of languages having a 1-round proof system and in this case refer to the prover as *Merlin* and the verifier as *Arthur*. In other words:

Definition 2 $L \in \mathsf{MA}$ if there exists a probabilistic polynomial-time verifier $\mathcal{V}$, a function s, and a polynomial p such that the following hold for all sufficiently long x:

- If $x \in L$ then there exists a string w (that can be sent by Merlin) such that

$$\Pr[\mathcal{V}(x, w) = 1] \geq s(|x|) + 1/p(|x|).$$

- If $x \notin L$ then for all w (sent by a cheating Merlin) it holds that

$$\Pr[\mathcal{V}(x, w) = 1] \leq s(|x|).$$

$\diamondsuit$

It is known that an equivalent definition is obtained even if we require perfect completeness and negligible soundness error.

2.1 Zero Knowledge Proof Systems

A *distribution ensemble* $\{X(a)\}_{a \in \{0,1\}^*}$ is an infinite sequence of probability distributions, where a distribution $X(a)$ is associated with each value of a. Two distribution ensembles $X = \{X(a)\}_{a \in \{0,1\}^*}$ and $Y = \{Y(a)\}_{a \in \{0,1\}^*}$ are *computationally indistinguishable* if for all probabilistic polynomial-time algorithms D, there exists a negligible function μ such that for every a we have

$$\left|\Pr[D(X(a), a) = 1] - \Pr[D(Y(a), a) = 1]\right| \leq \mu(|a|).$$

(We do not need to consider non-uniform distinguishers here since non-uniformity can be incorporated via the auxiliary input that we will provide to the cheating verifier, below.)

Given interactive algorithms $\mathcal{P}, \mathcal{V}^*$, we let $\mathsf{trans}_{\mathcal{V}^*}\langle \mathcal{P}(x), \mathcal{V}^*(y)\rangle$ denote the transcript of the indicated interaction; for convenience, this includes both the messages of the prover as well as those of the verifier. (We do not need to consider the entire *view* of $\mathcal{V}^*$ since we will restrict to deterministic verifiers, as justified below; note further that the input $y = (x, z)$ of $\mathcal{V}^*$ is provided to the distinguisher as per our definition of computational indistinguishability, above.) We now review the standard definitions for computational zero-knowledge proofs.

Definition 3 An interactive proof system $\mathcal{P}, \mathcal{V}$ for a language L is said to be a computational zero-knowledge proof system if for any probabilistic polynomial-time algorithm $\mathcal{V}^*$ there exists an expected polynomial-time simulator $\mathcal{S}$ such that the following distribution ensembles are computationally indistinguishable:

$$\{\mathsf{trans}_{\mathcal{V}^*}\langle \mathcal{P}(x), \mathcal{V}^*(x, z)\rangle\}_{x \in L, z \in \{0,1\}^*} \quad \text{and} \quad \{\mathcal{S}(x, z)\}_{x \in L, z \in \{0,1\}^*} .$$

$\diamondsuit$

The above definition incorporates an auxiliary input z provided to $\mathcal{V}^*$, and we may therefore restrict our consideration to cheating verifiers $\mathcal{V}^*$ that are deterministic (since we may view the randomness as being included in z). Note also that we allow simulation in *expected* polynomial time; this makes our results stronger. (Constant-round, black-box CZK proofs with strict polynomial-time simulation are already ruled out by Barak and Lindell [4].)

A computational zero-knowledge proof system $(\mathcal{P}, \mathcal{V})$ is *black-box* zero knowledge if there exists a "universal" simulator that takes oracle access to the cheating verifier $\mathcal{V}^*$. That is:

Definition 4 A computational zero-knowledge proof system $\mathcal{P}, \mathcal{V}$ is $\mathsf{black\text{-}box\ zero\text{-}knowledge}$ if there exists an expected polynomial-time oracle machine Sim (the $\mathsf{black\text{-}box\ simulator}$) such that for any probabilistic polynomial-time algorithm $\mathcal{V}^*$ the following distribution ensembles are computationally indistinguishable:

$$\{\mathsf{trans}_{\mathcal{V}^*}\langle \mathcal{P}(x), \mathcal{V}^*(x,z)\rangle\}_{x\in L, z\in\{0,1\}^*} \quad \text{and} \quad \left\{\mathsf{Sim}^{\mathcal{V}^*(x,z)}(x)\right\}_{x\in L, z\in\{0,1\}^*} .$$

$\diamondsuit$

We denote by ${}^{bb}\mathsf{CZK}(r)$ the class of languages that have r-round, *black-box*, computational zero-knowledge proof systems with negligible soundness error.

Terminology and simplifying assumptions. We will be concerned with 4-round CZK proof systems where (without loss of generality) the verifier sends the first message and the prover sends the final message. We use $\alpha, \beta, \gamma, \delta$ to denote the first, second, third, and fourth messages, respectively. $\mathcal{P}_x$ (resp., $\mathcal{V}_x$) denotes the honest prover (resp., honest verifier) algorithm when the common input is x.

We let $\alpha = \mathcal{V}_x(r)$ denote the first message sent by $\mathcal{V}_x$ when its random coins are fixed to r, and let $\gamma = \mathcal{V}_x(\alpha, \beta; r)$ denote the third-round message sent by $\mathcal{V}_x$ when it receives the second-round message β. Finally, $\mathcal{V}_x(\alpha, \beta, \gamma, \delta; r)$ is a bit denoting whether the verifier accepts (i.e., outputs 1) or rejects, given final message δ. We say that $(\alpha, \beta, \gamma, \delta, r)$ is an *accepting transcript* for a given x if $\mathcal{V}_x(\alpha, \beta, \gamma, \delta; r) = 1$. Note that we do not require the verifier's decision to depend on the actual transcript alone, but allow its decision to also possibly depend on its random coins.

Without loss of generality, we make a number of simplifying assumptions about the behavior of black-box simulator Sim. The first query of Sim to $\mathcal{V}^*$ will simply be a "prompt" query to which $\mathcal{V}^*$ responds with α. Subsequent queries by Sim are all of the form (α, β) (for some β of Sim's choice), to which $\mathcal{V}^*$ will respond with some γ. (We can assume Sim makes no queries of the form $(\alpha, \beta, \gamma, \delta)$ since $\mathcal{V}^*$ can simply refuse to respond to such queries.) We assume Sim makes any given query only once. Finally, if the simulator outputs the transcript $(\alpha', \beta, \gamma, \delta)$ we assume that $\alpha' = \alpha$, and that the simulator previously queried (α, β) to $\mathcal{V}^*$ and received response γ. This is without loss of generality since we can always force the simulator to make the query (α, β) (if it has not done so already) immediately before it outputs the transcript.

3 CZK Proof Systems with Perfect Completeness

We now state our main result:

Theorem 1 ${}^{bb}\mathsf{CZK}(4) \subseteq \mathsf{coMA}$.

In this section we prove this result in the easier case when the proof system in question has perfect completeness; we handle the case of imperfect completeness in the following section.

As intuition for the proof, consider first the case of a malicious verifier $\hat{\mathcal{V}}$ who acts in the following way: it sends an initial message α, and then in response to the prover's second message β it chooses a random message γ consistent with α. (For now, we ignore the fact that this does not necessarily represent a feasible polynomial-time strategy.) Formally, if we let R_α denote the set of random coins consistent with α (i.e., $r \in R_\alpha$ implies $\mathcal{V}_x(r) = \alpha$), then in response to β the malicious verifier chooses a random $r \in R_\alpha$ and computes $\gamma = \mathcal{V}_x(\alpha, \beta; r)$. Intuitively, it will be difficult to simulate an accepting transcript for such a verifier since each time the simulator "rewinds" $\hat{\mathcal{V}}$ it will be given a message γ consistent with a possibly *different* set of random coins. In fact, one can prove that if $x \notin L$ then the simulator will *not* be able to simulate an accepting transcript for such a verifier, since the ability to do so with non-negligible probability could be translated into the ability to violate the soundness condition of the proof system with non-negligible probability. (A proof of this fact goes along similar lines as the proof of the Goldreich-Krawczyk result [21].)

On the other hand, consider the case when $x \in L$. From the perspective of the honest prover, the behavior of $\hat{\mathcal{V}}$ is identical to that of the honest verifier, and so the honest prover's interaction with $\hat{\mathcal{V}}$ leads to an accepting transcript with probability 1. We would like to now use the zero-knowledge condition to show that Sim simulates an accepting transcript for such a verifier with high probability. Unfortunately, $\hat{\mathcal{V}}$ as described above may not run in polynomial time, whereas simulation is only guaranteed for polynomial-time verifiers.

It is possible, however, to obtain a polynomial-time cheating verifier with the desired behavior by providing the verifier as *auxiliary input* a sequence of sufficiently many coins $r_1, \ldots, r_s$ that are all consistent with the same first message α. Specifically, consider the verifier $\mathcal{V}^*$ defined as follows: given auxiliary input $r_1, \ldots, r_s$ (all consistent with the same first message α) and a poly-wise independent hash function h, send α as the first message. In response to the prover's second message β, compute $i = h(\beta)$ and use r_i to compute the next message $\gamma = \mathcal{V}_x(\alpha, \beta; r_i)$. Note that if $r_1, \ldots, r_s$ are chosen at random (subject to the constraint that they are all mutually consistent) then the behavior of $\mathcal{V}^*$ is identical to the behavior of $\hat{\mathcal{V}}$ as far as the honest prover is concerned. Since $\mathcal{V}^*$ runs in polynomial time, we are now able to argue that Sim simulates an accepting transcript for $\mathcal{V}^*$ with high probability when $x \in L$. Furthermore, it is still possible to show (using a slightly more complicated argument) that, with overwhelming probability, Sim *fails* to simulate an accepting transcript for this verifier whenever $x \notin L$.

Based on the above, we obtain an MA proof system for $\bar{L}$: on common input x, Merlin sends Arthur a sequence $r_1, \ldots, r_s$ of random coins that are all consistent with the same first message α, and Arthur runs $\mathsf{Sim}^{\mathcal{V}^*}(x)$. If this does *not* result in an accepting transcript then Arthur accepts, while if it does lead to an accepting transcript then Arthur rejects. In what follows, we formalize the above intuition and show how to handle various technicalities that arise.

3.1 Technical Details

Fix $L \in {}^{bb}\mathsf{CZK}(4)$. This means that, for this language, there exists a prover $\mathcal{P}$, a verifier $\mathcal{V}$, and a black-box simulator Sim satisfying Definitions 1–4 (except that, in this section, we assume perfect completeness). Assume without loss of generality that the second message of the protocol (on input x) always has length $m(|x|)$, and let $\ell(|x|)$ denote the number of random coins used by $\mathcal{V}$. Let $T(|x|)$ denote a polynomial upper-bound on the expected running time of Sim.

Consider the following MA proof system for the language $\bar{L}$, where Merlin (i.e., the prover) and Arthur (i.e., the verifier) share in advance an input x of length n:

Notation: Let $\ell = \ell(n)$, $m = m(n)$, and $T = T(n)$. Set $s = 50 \cdot T^2$; note that s is polynomial in n.

Merlin's message: Merlin sends a sequence of s coins $r_1, \ldots, r_s \in \{0,1\}^\ell$. (For the honest Merlin, these are all consistent with the same first message α. See the proof of Claim 2 for details.)

Arthur's actions: Arthur proceeds as follows:

1. Set $\alpha = \mathcal{V}_x(r_1)$. Check that $\alpha = \mathcal{V}_x(r_i)$ for all $1 < i \leq s$, i.e., that all the random coins are consistent with the same first message α. If not, reject; otherwise, go to the next step.
2. Choose a random $5T$-wise independent hash function $h : \{0,1\}^m \to \{1, \ldots, s\}$. Construct the following deterministic verifier $\mathcal{V}^*_\alpha$:
   (a) Send first message α to the prover.
   (b) Upon receiving message β from the prover, compute $i = h(\beta)$ and send the message $\gamma = \mathcal{V}_x(\alpha, \beta; r_i)$ to the prover.
3. Run $\mathsf{Sim}^{\mathcal{V}^*_\alpha}(x)$ for at most $5T$ steps using uniform random coins for Sim. If Sim does *not* output an accepting transcript within this time bound, output "accept". Otherwise, output "reject". (Formally, output "reject" iff Sim outputs $(\alpha, \beta, \gamma, \delta)$, within the allotted time bound, such that $\mathcal{V}_x(\alpha, \beta, \gamma, \delta; r_{h(\beta)}) = 1$.)

The following claims show that the above is a valid MA-protocol for $\bar{L}$, thus proving Theorem 1 for the case of protocols having perfect completeness.

Claim 1 *For sufficiently long $x \notin \bar{L}$ and for any message $r_1, \ldots, r_s$ sent by Merlin, the probability that Arthur accepts is at most* $2/5$.

Proof Fix some $r_1, \ldots, r_s$ sent by Merlin. Set $\alpha = \mathcal{V}_x(r_1)$, and assume $\mathcal{V}_x(r_i) = \alpha$ for all $1 \leq i \leq s$ since, if not, Arthur rejects immediately. Define $\mathcal{V}^*_\alpha$ as in the description of Arthur. When $x \notin \bar{L}$ we have $x \in L$ and, by perfect completeness, the interaction of the honest prover $\mathcal{P}_x$ with $\mathcal{V}^*_\alpha$ would result in an accepting transcript with probability 1. (To see this, note that an execution of $\mathcal{V}^*_\alpha$ is equivalent to an execution of the honest verifier $\mathcal{V}_x$ using random coins $r_{h(\beta)}$.) The zero-knowledge condition thus implies that, for x sufficiently long, $\mathsf{Sim}^{\mathcal{V}^*_\alpha}(x)$ outputs an accepting conversation with probability at least $1 - \mathsf{negl}(n) > 4/5$. It follows that even the truncated version of Sim, where its execution is halted after $5T$ steps, outputs an accepting conversation with probability at least $3/5$. Arthur thus accepts with probability at most $2/5$, as claimed. ■

Claim 2 *For sufficiently long $x \in \bar{L}$, there exists a message $r_1, \ldots, r_s$ such that Arthur will accept with probability at least* $1/2$.

Proof Fix $x \in \bar{L}$. We show a randomized strategy that allows Merlin to convince Arthur with probability at least $1/2$; this implies the claim.

Merlin proceeds as follows: choose random $r_1 \in \{0,1\}^\ell$ and compute $\alpha = \mathcal{V}_x(r_1)$. Define $R_\alpha \overset{\text{def}}{=} \{r \mid \mathcal{V}_x(r) = \alpha\}$; i.e., R_α is the set of coins for the honest verifier consistent with the first message α. Then choose $r_2, \ldots, r_s$ uniformly from R_α. (These need not be distinct.) Send

$r_1, \ldots, r_s$ to Arthur. Let p^* denote the probability that Arthur rejects. Note that this is exactly the probability that $\mathsf{Sim}^{\mathcal{V}^*_\alpha}(x)$ outputs an accepting transcript within the allotted time bound.

We upper-bound p^* by considering a slightly different experiment involving an all-powerful cheating prover $\mathcal{P}^*$ attempting to falsely convince the honest verifier $\mathcal{V}_x$ that $x \in L$. The strategy of $\mathcal{P}^*$ is defined as follows:

1. Receive message α from the real verifier. Let $R_\alpha \stackrel{\text{def}}{=} \{r \mid \mathcal{V}_x(r) = \alpha\}$.

2. Run Sim using uniform random coins, for at most $5T$ steps. Sim expects to be given oracle access to a (possibly cheating) verifier, and $\mathcal{P}^*$ simulates the actions of such a verifier by choosing a random index $q \leftarrow \{1, \ldots, 5T\}$ and then proceeding as follows:

   (a) Send α as the verifier's first message.

   (b) In response to the i^{th} simulator message (α, β_i) for $i \neq q$, choose a random $r_i \leftarrow R_\alpha$, compute $\gamma_i = \mathcal{V}_x(\alpha, \beta_i; r_i)$, and give γ_i to Sim. (Recall we assume that Sim never makes the same query twice.)

   (c) In response to the q^{th} simulator message (α, β_q), send β_q to the (external) honest verifier, and receive in return a message γ_q. Give γ_q to Sim.

3. If Sim outputs a conversation $(\alpha, \beta, \gamma, \delta)$ with $\beta = \beta_q$ within the allotted time bound, then send δ to the (external) honest verifier.

In the above experiment, each "query" β_i of Sim is answered by using a random element $r_i \leftarrow R_\alpha$ to compute the response $\gamma_i = \mathcal{V}_x(\alpha, \beta_i; r_i)$. This is immediate for $i \neq q$, but is true also for $i = q$ since, from the perspective of $\mathcal{P}^*$ and Sim, the coins being used by the external, honest verifier are uniformly distributed in R_α. Let $\hat{p}$ denote the probability that Sim outputs an accepting transcript in this case, within the allotted time bound. Since Sim makes at most $5T$ queries to its oracle in the above experiment, $\mathcal{P}^*$ convinces the honest verifier to accept with probability $\hat{p}/5T$. Since the proof system has negligible soundness error we have that, for x sufficiently long, $\hat{p} \leq 1/4$.

We return now to consideration of p^*. When Arthur runs $\mathsf{Sim}^{\mathcal{V}^*_\alpha}(x)$, he does so by first choosing a random h and then answering the simulator's i^{th} query (α, β_i) by using $r_{h(\beta_i)}$ to compute the response $\gamma_i = \mathcal{V}_x(\alpha, \beta_i; r_{h(\beta_i)})$. Since Merlin chooses each of the r_i uniformly from R_α, these responses are distributed identically to the above experiment unless there is a collision in h; that is, unless there exist some $\beta_i \neq \beta_j$ with $h(\beta_i) = h(\beta_j)$. Because h is chosen in a $5T$-wise independent fashion and Sim is restricted to making only $5T$ queries, a standard birthday bound shows that the probability of such a collision is at most $(5T)^2/2s = 1/4$. Conditioned on a collision not occurring, the probability that $\mathsf{Sim}^{\mathcal{V}^*_\alpha}(x)$ outputs an accepting conversation is exactly $\hat{p} \leq 1/4$. We conclude that $p^* \leq 1/4 + 1/4 = 1/2$, and so Arthur rejects with probability at most $1/2$ (and accepts with probability at least $1/2$). ■

4 Handling Imperfect Completeness

In the previous section we assumed perfect completeness, and this assumption is essential for the MA proof system given there. To see the problem, assume $\mathcal{P}, \mathcal{V}$ is such that the honest verifier always rejects whenever its random coins are all 0. Then a cheating Merlin can send $r_1 = \cdots = r_s = 0^\ell$ and this will cause Arthur to accept with probability 1 even when $x \notin \bar{L}$.

In the modified MA proof system we describe in this section, we address the problem raised above by having Arthur "verify" that Merlin sends "representative" random coins $r_1, \ldots, r_s$. We do this by having Arthur check that $\mathsf{Sim}^{\mathcal{V}_x(r_i)}(x)$, for a random index $i \in \{1, \ldots, s\}$, outputs an accepting transcript with "high" probability. Arthur rejects immediately if this is not the case; otherwise, Arthur accepts if $\mathsf{Sim}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x)$ *fails* to output an accepting transcript (as in the previous section). Unfortunately, this may cause a problem with completeness (i.e., for the honest Merlin when $x \in \bar{L}$): there may be instances $x \in \bar{L}$ for which $\mathsf{Sim}^{\mathcal{V}_x(r_i)}(x)$ *never* outputs an accepting transcript. (E.g., if x is such that the verifier $\mathcal{V}$ in the underlying proof system can decide on its own that $x \notin L$ without any help from $\mathcal{P}$.) We therefore add an additional test at the beginning of the MA protocol: Arthur runs $\mathsf{Sim}^{\mathcal{V}_x(r)}(x)$ using random coins r that it chooses itself, and accepts that $x \in \bar{L}$ if this execution of Sim fails to output an accepting transcript. (This is only the intuition behind our protocol; the technical details are slightly different.)

Before presenting the modified Arthur-Merlin protocol, we introduce some notation. For a given randomized experiment Expt that depends on random coins r, we let $\mathsf{estimate}_\varepsilon(\Pr_r[\mathsf{Expt} = 1])$ denote a procedure that outputs an estimate to the given probability (taken over randomness r) to within an additive factor of ε, except with probability at most ε. That is:

$$\Pr\left[\,\left|\mathsf{estimate}_\varepsilon(\Pr_r[\mathsf{Expt} = 1]) - \Pr_r[\mathsf{Expt} = 1]\right| \geq \varepsilon\right] \leq \varepsilon.$$

This can be done in the standard way using $\Theta(\varepsilon^{-2} \log \frac{1}{\varepsilon})$ independent executions of Expt. The important thing to note is that when ε is noticeable (and Expt can be run in polynomial time), this estimation can be done in polynomial time. In the experiments we will be considering, some variables will be fixed as part of the experiment and others will be chosen at random; we will always subscript those variables being chosen at random (as done above with the subscripted r).

Below, we let $\mathcal{V}^*$ denote the same malicious verifier as in the previous section. Specifically, on input x and auxiliary input $z = r_1, \ldots, r_s, h$, where each r_i represents coins for the honest verifier and h is a hash function, $\mathcal{V}^*$ acts as follows:

1. Send first message $\alpha = \mathcal{V}_x(r_1)$ to the prover.
2. Upon receiving message β from the prover, compute $i = h(\beta)$ and send the message $\gamma = \mathcal{V}_x(\alpha, \beta; r_i)$ to the prover.
3. Receive final message δ from the prover.

We say an interaction of $\mathcal{P}_x$ with $\mathcal{V}^*(x, z)$ *results in an accepting transcript* if $(\alpha, \beta, \gamma, \delta, r_i)$ is an accepting transcript.

Let $L \in {}^{bb}\mathsf{CZK}(4)$, and say L has a 4-round CZK proof system $\mathcal{P}, \mathcal{V}$ with acceptance probability $c(\cdot)$ where c is noticeable (i.e., $c = \Omega(1/p(\cdot))$ for some polynomial p). Let Sim be the black-box simulator for the proof system, and let ℓ, m, and T be as in the previous section.

Once again, Merlin and Arthur share in advance an input x of length n. The MA proof system for the language $\bar{L}$ follows:

Notation: Let $c = c(n)$, $\ell = \ell(n)$, $m = m(n)$, and $T = T(n)$. Assume n is large enough so that $c > 0$. Set $\varepsilon = c/20$, and $s = 4T^2\varepsilon^{-3}$. (Note that ε is noticeable, and s is polynomial.) Let $\widetilde{\mathsf{Sim}}$ denote an execution of Sim for at most $2T/\varepsilon$ steps, and making exactly $2T/\varepsilon$ queries.

Merlin's message: Merlin sends a sequence of s coins $r_1, \ldots, r_s \in \{0, 1\}^\ell$.

Arthur's actions: Arthur proceeds as follows:

1. Compute
$$p_1 = \mathsf{estimate}_\varepsilon\left(\Pr_{r,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}_x(r)}(x;\rho) \text{ outputs an accepting transcript}\right]\right).$$
If $p_1 < c - 2\varepsilon$ then accept; otherwise, continue to the next step.

2. Set $\alpha = \mathcal{V}_x(r_1)$. Check that $\alpha = \mathcal{V}_x(r_i)$ for all $1 < i \leq s$. If not, reject; otherwise, continue to the next step.

3. Choose $i \leftarrow \{1, \ldots, s\}$ and coins ρ and run $\widetilde{\mathsf{Sim}}^{\mathcal{V}_x(r_i)}(x;\rho)$. If this does not result in an accepting transcript, reject; otherwise, continue to the next step.

4. Let H denote a family of $2T/\varepsilon$-wise independent hash functions $h : \{0,1\}^m \to \{1, \ldots, s\}$. Compute
$$p_2 = \mathsf{estimate}_\varepsilon\left(\Pr_{h \leftarrow H,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho) \text{ outputs an accepting transcript}\right]\right).$$
If $p_2 < c - 10\varepsilon$ accept; else reject.

(It should be clear that we have not attempted to optimize any of the parameters of the above proof system.) We now prove claims analogous to those in the previous section.

Claim 3 *For any $x \notin \bar{L}$ sufficiently long and for any message $r_1, \ldots, r_s$ sent by Merlin, the probability that Arthur accepts is at most $c - 6\varepsilon$.*

Proof If $x \notin \bar{L}$ then $x \in L$ and so the interaction of $\mathcal{P}_x$ with $\mathcal{V}_x$ results in an accepting transcript with probability at least c. The zero-knowledge condition implies that, for x sufficiently long,
$$\Pr_{r,\rho}[\widetilde{\mathsf{Sim}}^{\mathcal{V}_x(r)}(x;\rho) \text{ outputs an accepting transcript}] \geq c - \varepsilon.$$
This means that, except with probability at most ε, the value p_1 computed by Arthur satisfies $p_1 \geq c - 2\varepsilon$; thus, Arthur accepts in the first step with probability at most ε.

Fix some $r_1, \ldots, r_s$ sent by Merlin. We may assume $\mathcal{V}_x(r_i) = \mathcal{V}_x(r_j)$ for all $1 \leq i, j \leq s$ since, if not, Arthur rejects in the second step. Define
$$\hat{p} = \Pr_{i \leftarrow \{1,\ldots,s\},\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}_x(r_i)}(x;\rho) \text{ outputs an accepting transcript}\right].$$
There are two cases to consider:

Case 1: If $\hat{p} < c - 7\varepsilon$, then the probability that Arthur does not reject in step 3 is at most $c - 7\varepsilon$.

Case 2: On the other hand, if $\hat{p} \geq c - 7\varepsilon$ then (again using the zero-knowledge property)
$$\Pr_{i \leftarrow \{1,\ldots,s\},r}\left[\langle \mathcal{P}_x(r), \mathcal{V}_x(r_i)\rangle = 1\right] \geq c - 8\varepsilon.$$
By definition of $\mathcal{V}^*$ it holds that
$$\Pr_{h \leftarrow H,r}\left[\langle \mathcal{P}_x(r), \mathcal{V}^*(x, r_1, \ldots, r_s, h)\rangle \text{ results in an accepting transcript}\right]$$
$$= \Pr_{i \leftarrow \{1,\ldots,s\},r}\left[\langle \mathcal{P}_x(r), \mathcal{V}_x(r_i)\rangle = 1\right].$$

Thus, relying on the zero-knowledge property once again,

$$\mathrm{Pr}_{h\leftarrow H,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho)\text{ outputs an accepting transcript}\right]\geq c-9\varepsilon.$$

So, except with probability at most ε, the value p_2 computed by Arthur satisfies $p_2 \geq c - 10\varepsilon$; thus, Arthur accepts in the last step with probability at most ε.

Combining the above, we see that Arthur accepts with probability at most $\varepsilon + \max\{c - 7\varepsilon, \varepsilon\}$, which is at most $c - 6\varepsilon$. ■

Claim 4 *For any $x \in \bar{L}$ sufficiently long, there exists a message $r_1, \ldots, r_s$ such that Arthur will accept with probability at least $c - 5\varepsilon$.*

Proof Fix $x \in \bar{L}$. Define

$$\hat{p} = \mathrm{Pr}_{r,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}_x(r)}(x;\rho)\text{ outputs an accepting transcript}\right].$$

There are two cases to consider:

Case 1: If $\hat{p} < c - 3\varepsilon$ then, except with probability at most ε, the value p_1 computed by Arthur satisfies $p_1 < c - 2\varepsilon$; thus, Arthur accepts in the first step with probability at least $1 - \varepsilon \geq c - 5\varepsilon$.

Case 2: On the other hand, say $\hat{p} \geq c - 3\varepsilon$. As in the proof of Claim 2, Merlin proceeds as follows: choose random $r_1 \in \{0,1\}^\ell$ and compute $\alpha = \mathcal{V}_x(r_1)$. Let $R_\alpha \stackrel{\text{def}}{=} \{r \mid \mathcal{V}_x(r) = \alpha\}$, and choose $r_2, \ldots, r_s$ uniformly from R_α. Send $r_1, \ldots, r_s$ to Arthur. We show that Arthur will accept with high probability, taken over its own coins and Merlin's message.

Arthur can reject in either step 3 or step 4. We upper-bound the probability that Arthur rejects in either of these steps individually, and then apply a union bound to upper-bound the total probability that Arthur rejects.

Each r_i, taken individually, is uniformly distributed in $\{0,1\}^\ell$. Thus, in step 3, choosing a random $i \in \{1, \ldots, s\}$ and using coins r_i is equivalent to choosing uniformly random coins for $\mathcal{V}_x$. It follows that the probability that Arthur rejects in step 3 is exactly equal to $1 - \hat{p} \leq 1 - c + 3\varepsilon$.

We proceed to analyze step 4. As in the proof of Claim 2, say a *collision* occurs in an execution of $\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho)$ if the simulator makes two distinct queries (α, β_i) and (α, β_j) for which $h(\beta_i) = h(\beta_j)$. Let coll denote such an event. As before, we have

$$\Pr_{r_1,\ldots,r_s,h,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho)\text{ outputs an accepting transcript}\right]\leq \tag{1}$$
$$\Pr_{r_1,\ldots,r_s,h,\rho}[\mathsf{coll}]+\Pr_{r_1,\ldots,r_s,h,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho)\text{ outputs an accepting transcript}\mid\overline{\mathsf{coll}}\right],$$

where $r_1, \ldots, r_s$ are chosen by Merlin as described above (and not uniformly and independently at random). The probability of a collision is independent of $r_1, \ldots, r_s$, and is upper-bounded by $\Pr[\mathsf{coll}] \leq \frac{(2T/\varepsilon)^2}{2s} = \frac{\varepsilon}{2}$. As in the proof of Claim 2, for sufficiently long x it holds that

$$\mathrm{Pr}_{r_1,\ldots,r_s,h,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho)\text{ outputs an accepting transcript}\mid\overline{\mathsf{coll}}\right]\leq\varepsilon^2/2;$$

this means that, except with probability at most ε, the $r_1, \ldots, r_s$ chosen by Merlin satisfy

$$\Pr_{h,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho) \text{ outputs an accepting transcript} \mid \overline{\mathsf{coll}}\right] \le \varepsilon/2.$$

Combined with Equation (1), this means that with probability at most ε the $r_1, \ldots, r_s$ chosen by Merlin satisfy

$$\Pr_{h,\rho}\left[\widetilde{\mathsf{Sim}}^{\mathcal{V}^*(x;r_1,\ldots,r_s,h)}(x;\rho) \text{ outputs an accepting transcript}\right] \le \varepsilon < c - 11\varepsilon.$$

Assuming the above to be the case, Arthur will reject in step 4 with probability at most ε. Taken together, this means that Arthur rejects in step 4 with probability at most 2ε.

Summing the probabilities of rejection in steps 3 and 4, we see that, overall, Arthur rejects with probability at most $1 - c + 5\varepsilon$, or accepts with probability at least $c - 5\varepsilon$. ■

5 Future Directions

Coupled with the obvious fact that ${}^{bb}\mathsf{CZK}(4) \subseteq \mathsf{AM}$, this work shows that ${}^{bb}\mathsf{CZK}(4) \subseteq \mathsf{AM} \cap \mathsf{coMA}$. Due to the similarity with the fact that $\mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [19, 1], as well as the fact that the only languages known to be in ${}^{bb}\mathsf{CZK}(4)$ (under any assumption) are also in SZK, it is natural to conjecture that ${}^{bb}\mathsf{CZK}(4) \subseteq \mathsf{SZK}$.

Another interesting direction would be to show any broad positive results for ${}^{bb}\mathsf{CZK}(4)$: say, proving that $\mathsf{NP} \cap \mathsf{coNP} \subseteq {}^{bb}\mathsf{CZK}(4)$.

Finally, is it possible to extend the techniques from [28] to show that there are no black-box constructions of *constant-round* (black-box) zero-knowledge proofs for NP based on one-way functions? Some recent progress on this question is reported in [36, 26]

Acknowledgments

I thank Hoeteck Wee for many illuminating discussions regarding the round complexity of black-box zero knowledge, and for helpful remarks regarding a previous draft of this paper. Thanks also to Dov Gordon and Arkady Yerukhimovich for reading and commenting on a preliminary version of this manuscript. Suggestions of various referees also helped to clarify the presentation.

References

[1] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Computer and System Sciences*, 42(3):327–345, 1991.

[2] L. Babai and S. Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *J. Computer and System Sciences*, 36(2):254–276, 1988.

[3] B. Barak. How to go beyond the black-box simulation barrier. In *Proc. 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–115. IEEE, 2001.

[4] B. Barak and Y. Lindell. Strict polynomial-time in simulation and extraction. *SIAM J. Computing*, 33(4):738–818, 2004.

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

[6] B. Barak, S.J. Ong, and S.P. Vadhan. Derandomization in cryptography. *SIAM J. Computing*, 37(2):380–400, 2007.

[7] M. Bellare, M. Jakobsson, and M. Yung. Round-optimal zero-knowledge arguments based on any one-way function. In *Advances in Cryptology — Eurocrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 280–305. Springer-Verlag, 1997.

[8] M. Bellare, S. Micali, and R. Ostrovsky. Perfect zero knowledge in constant rounds. In *Proc. 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 482–493. ACM, 1990.

[9] M. Bellare, S. Micali, and R. Ostrovsky. The (true) complexity of statistical zero knowledge. In *Proc. 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 494–502. ACM, 1990.

[10] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Advances in Cryptology — Crypto 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289. Springer-Verlag, 2004.

[11] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero knowledge. In *Advances in Cryptology — Crypto '88*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer-Verlag, 1990.

[12] R. Boppana, J. Håstad, and S. Zachos. Does coNP have short interactive proofs? *Information Proc. Letters*, 25(2):127–132, 1987.

[13] J. Boyar, S. Kurtz, and M. Krentel. Discrete logarithm implementation of perfect zero-knowledge blobs. *J. Cryptology*, 2(2):63–76, 1990.

[14] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *J. Computer and Systems Sciences*, 37(2):156–189, 1988.

[15] R. Cramer, I. Damgård, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *Public-Key Cryptography (PKC) 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–372. Springer-Verlag, 2000.

[16] G. Di Crescenzo and G. Persiano. Round-optimal perfect zero-knowledge proofs. *Information Proc. Letters*, 50(2):93–99, 1994.

[17] I. Damgård, M. Pedersen, and B. Pfitzmann. On the existence of statistically-hiding bit commitment schemes and fail-stop signatures. *J. Cryptology*, 10(3):163–194, 1997.

[18] U. Feige and A. Shamir. Zero knowledge proofs of knowledge in two rounds. In *Advances in Cryptology — Crypto '89*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544. Springer-Verlag, 1990.

[19] L. Fortnow. The complexity of perfect zero knowledge. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.

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

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

[22] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity, or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[23] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

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

[25] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, 1988.

[26] S.D. Gordon, H. Wee, D. Xiao, and A. Yerukhimovich. On the round complexity of zero-knowledge proofs based on one-way permutations. In *Progress in Cryptology — Latincrypt 2010*, volume 6212 of *Lecture Notes in Computer Science*, pages 189–204. Springer-Verlag, 2010.

[27] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In *Advances in Cryptology — Crypto '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423. Springer-Verlag, 1998. Available at `http://eprint.iacr.org/1999/009`.

[28] I. Haitner, J.J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols — a tight bound on the round complexity of statistically-hiding commitments. In *Proc. 48th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 669–679. IEEE, 2007. Available at `http://eprint.iacr.org/2007/145`.

[29] I. Haitner, M.-H. Nguyen, S.J. Ong, O. Reingold, and S.P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Computing*, 39(3):1153–1218, 2009.

[30] S. Halevi and S. Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology — Crypto '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 201–215. Springer-Verlag, 1996.

[31] R. Impagliazzo and M. Yung. Direct minimum-knowledge computations. In *Advances in Cryptology — Crypto '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer-Verlag, 1988.

[32] T. Itoh and K. Sakurai. On the complexity of constant-round ZKIP of possession of knowledge. In *Advances in Cryptology — Asiacrypt '91*, volume 739 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, 1993.

[33] K. Kurosawa, W. Ogata, and S. Tsujii. 4-move perfect ZKIP for some promise problems. *IEICE Trans. on Fundamentals of Electronics, Communications, and Computer Sciences*, E78-A(1):34–41, 1995.

[34] M. Lepinski. On the existence of 3-round zero-knowledge proofs. Master's thesis, MIT, 2002. Available at `http://theory.lcs.mit.edu/~cis/cis-theses.html`.

[35] S. J. Ong and S. Vadhan. An equivalence between zero knowledge and commitments. In *3rd Theory of Cryptography Conference (TCC)*, volume 4948 of *Lecture Notes in Computer Science*, pages 482–500. Springer-Verlag, 2008.

[36] R. Pass and M. Venkitasubramaniam. Private coins versus public coins in zero-knowledge proof systems. In *5th Theory of Cryptography Conference (TCC)*, volume 5978 of *Lecture Notes in Computer Science*, pages 588–605. Springer-Verlag, 2010.

[37] T. Saito, K. Kurosawa, and K. Sakurai. 4-move perfect SKIP of knowledge with no assumption. In *Advances in Cryptology — Asiacrypt '91*, volume 739 of *Lecture Notes in Computer Science*, pages 320–331. Springer-Verlag, 1993.

[38] S. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, MIT, 1999.