Can Statistical Zero Knowledge be Made Non-interactive? or On the Relationship of $\mathcal{SZK}$ and $\mathcal{NISZK}$*

[Extended Abstract]

Oded Goldreich[1], Amit Sahai[2], and Salil Vadhan[2]

[1] Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel. E-mail: oded@wisdom.weizmann.ac.il. Work done while visiting LCS, MIT.

[2] Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: {amits,salil}@theory.lcs.mit.edu. URL: http://theory.lcs.mit.edu/~salil.

Abstract. We extend the study of non-interactive statistical zero-knowledge proofs. Our main focus is to compare the class $\mathcal{NISZK}$ of problems possessing such *non-interactive* proofs to the class $\mathcal{SZK}$ of problems possessing *interactive* statistical zero-knowledge proofs. Along these lines, we first show that if statistical zero knowledge is non-trivial then so is non-interactive statistical zero knowledge, where by non-trivial we mean that the class includes problems which are *not* solvable in probabilistic polynomial-time. (The hypothesis holds under various assumptions, such as the intractability of the Discrete Logarithm Problem.) Furthermore, we show that if $\mathcal{NISZK}$ is closed under complement, then in fact $\mathcal{SZK} = \mathcal{NISZK}$, i.e. all statistical zero-knowledge proofs can be made non-interactive.

The main tools in our analysis are two promise problems that are natural restrictions of promise problems known to be complete for $\mathcal{SZK}$. We show that these restricted problems are in fact complete for $\mathcal{NISZK}$ and use this relationship to derive our results comparing the two classes. The two problems refer to the statistical difference, and difference in entropy, respectively, of a given distribution from the uniform one. We also consider a weak form of $\mathcal{NISZK}$, in which only requires that for every inverse polynomial $1/p(n)$, there exists a simulator which achieves simulator deviation $1/p(n)$, and show that this weak form of $\mathcal{NISZK}$ actually equals $\mathcal{NISZK}$.

Keywords: Statistical Zero-Knowledge Proofs, Non-Interactive Zero-Knowledge Proofs.

* The full version of this paper has been submitted to the *Electronic Colloquium on Computational Complexity* [26].

Michael Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 467–484, 1999.

1 Introduction

Zero-Knowledge proofs, introduced by Goldwasser, Micali and Rackoff [29], are fascinating and extremely useful constructs. Their fascinating nature is due to their seemingly contradictory nature; they are both convincing and yet yield nothing beyond the validity of the assertion being proven. Their applicability in the domain of cryptography is vast; they are typically used to force malicious parties to behave according to a predetermined protocol (which requires parties to provide proofs of the correctness of their secret-based actions without revealing these secrets). Zero-knowledge proofs come in many flavors, and in this paper we focus on two parameters: The first parameter is the underlying *communication model*, and the second is the *type of the zero-knowledge guarantee.*

The communication model. When Goldwasser, Micali, and Rackoff proposed the definition of zero-knowledge proofs, it seemed that interaction was crucial to achieving zero knowledge – that the possibility of zero knowledge arose through the power of interaction. Indeed, it was not unexpected when [24] showed zero knowledge to be trivial (i.e., only exists for proofs of $\mathcal{BPP}$ statements) in the most straightforward non-interactive models. Surprisingly, however, Blum, Feldman, and Micali [7], showed that by changing the model slightly, it is possible to achieve zero knowledge in a non-interactive setting (i.e. where only unidirectional communication can occur). Specifically, they assume that both Prover and Verifier have access to a shared truly random string, called the *reference string.* Aside from this assumption, all communication consists of one message, the "proof," which is generated by the Prover (based on the assertion being proved and the reference string) and sent from the Prover to the Verifier.

Non-interactive zero-knowledge proofs, on top of being more communication-efficient by definition, have several applications not offered by ordinary interactive zero-knowledge proofs. They have been used, among other things, to build digital signature schemes secure against adaptive chosen message attack [3], public-key cryptosystems secure against chosen-ciphertext attack [34, 18], and non-malleable cryptosystems [18].

The zero-knowledge guarantee. For ordinary *interactive* zero-knowledge proofs, the zero-knoweldege requirement is formulated by saying that the transcript of the Verifier's interaction with the Prover can be *simulated* by the Verifier itself. Similarly, for the *non-interactive* setting described above, the zero-knowledge condition is formulated by requiring that one can produce, knowing only the statement of the assertion, a random reference string *along with* a "proof" that works for the reference string. More precisely, we require that there exists an efficient procedure that on input a valid assertion produces a distribution which is "similar" to the joint distribution of random reference strings and proofs generated by the Prover. The key parameter is the interpretation of "similarity." Two notions have been commonly considered in the literature (cf., [29, 23, 21, 6, 5]). *Statistical zero knowledge* requires that these distributions be statistically close (i.e., the statistical difference between them is negligible). *Computational zero* *knowledge* instead requires that these distributions are computationally indistinguishable (cf., [28, 41]). In this work, we focus on the stronger security requirement of statistical zero knowledge.

Since its introduction in [7], most work on non-interactive zero knowledge has focused on the computational type (cf., [7, 15, 16, 6, 20, 31]). With non-interactive statistical zero knowledge, the main objects of investigation have been the specific proof system for `Quadratic Nonresiduosity` and variants [6, 14, 11].[1] Recently, De Santis *et. al.* [12] opened the door to a general study of non-interactive statistical zero-knowledge by showing that it contains a complete (promise[2]) problem.

Notation. Throughout the paper, $\mathcal{SZK}$ denotes the class of promise problems having statistical zero-knowledge interactive proof systems (defined in Appendix A), and $\mathcal{NISZK}$ denotes the class of promise problems having non-interactive statistical zero-knowledge proof systems (defined in Section 1.1).

Our Contribution. In this work, we seek to understand what, if any, additional power interaction gives in the context of statistical zero knowledge. Thus, we continue the investigation of $\mathcal{NISZK}$, focusing on its relationship with $\mathcal{SZK}$. Our first result is that the non-triviality of $\mathcal{SZK}$ implies non-triviality of $\mathcal{NISZK}$, where by non-trivial we mean that a class includes problems which are *not* solvable in probabilistic polynomial-time. The hypothesis holds under various assumptions, such as the intractability of Discrete Logarithm Problem [22] (or Quadratic Residuosity [29] or Graph Isomorphism [23]), but variants of these last two problems are already known to be in $\mathcal{NISZK}$ [6, 5]).

Furthermore, we show that if $\mathcal{NISZK}$ is closed under complement, then in fact $\mathcal{SZK} = \mathcal{NISZK}$ — i.e., all statistical zero-knowledge proofs can be made non-interactive. We note that [12] have claimed that $\mathcal{NISZK}$ is closed under complement (and OR), but these claims have been retracted [13].

We also show the equivalence of $\mathcal{NISZK}$ with a variant in which the statistical zero knowledge requirement is weakened somewhat.

Complete Problems. Central to our methodology is the use of simple and natural complete problems to understand classes, such as $\mathcal{SZK}$ and $\mathcal{NISZK}$, whose definitions are rather complicated. In particular, we exhibit two natural promise problems and prove that they are complete for $\mathcal{NISZK}$. The two problems refer to the "distance" (in two different senses) of a given distribution from the uniform one. These two problems are natural restrictions of two promise problems shown complete for $\mathcal{SZK}$, in [38] and [27], respectively. Indeed, our results about the

[1] The only exception is an unpublished manuscript of Bellare and Rogaway [5] who proved some basic results about non-interactive perfect zero-knowledge and showed a non-interactive perfect zero-knowledge proof for the language of graphs with trivial automorphism group.

[2] A *promise problem* Π is a pair $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ of disjoint sets of strings, corresponding to YES and NO instances of a decision problem.

relationship between $\mathcal{SZK}$ and $\mathcal{NISZK}$ come from relating the corresponding complete problems. This general theme of using completeness to simplify the study of a class, rather than as evidence for computational intractability (as is the traditional use of $\mathcal{NP}$-completeness), has been evidenced in a number of recent works (cf., [23, 33, 40, 1, 2]) and has been particularly useful in understanding statistical zero knowledge (cf., [38, 39, 12, 27]).

1.1 The Non-interactive Model

Let us recall the definition of a non-interactive statistical zero-knowledge proof system from [6].[3] We will adapt the definition to promise problems. Note that our definition will capture what [6] call a *bounded proof system*, in that each shared reference string can only be used once. In contrast to non-interactive *computational* zero knowledge (cf., [6, 20]), it is unknown whether every problem that has such a (bounded) non-interactive *statistical* zero-knowledge proof system also has one in which the shared reference string can be used an unbounded (polynomial) number of times.

A *non-interactive statistical zero-knowledge* proof system for a promise problem Π is defined by a triple of probabilistic machines P, V, and S, where V and S are polynomial-time and P is computationally unbounded, and a polynomial $r(n)$ (which will give the size of the random *reference string* σ), such that:

1. (Completeness:) For all $x \in \Pi_{\mathrm{YES}}$, the probability that $V(x, \sigma, P(x, \sigma))$ accepts is at least 2/3.
2. (Soundness:) For all $x \in \Pi_{\mathrm{NO}}$, the probability that $V(x, \sigma, P(x, \sigma))$ accepts is at most 1/3.
3. (Zero Knowledge:) For all $x \in \Pi_{\mathrm{YES}}$, the statistical deviation between the following two distributions is at most $\beta(|x|)$:
   (A) Choose σ uniformly from $\{0,1\}^{r(|x|)}$, sample p from $P(x, \sigma)$, and output (p, σ).
   (B) $S(x)$ (where the coins for S are chosen uniformly at random.)

where $\beta(n)$ is a negligible function,[4] termed the *simulator deviation*, and the probabilities in Conditions 1 and 2 are taken over the random coins of V and P, and the choice of σ uniformly from $\{0,1\}^{r(n)}$. Note that non-interactive statistical zero knowledge is closed under parallel repetition, so the completeness and soundness errors (i.e. the probability of rejection (resp., acceptance) for YES (resp., NO) instances) can be made exponentially small in $|x|$.

We also define a weaker notion of zero knowledge, known as a *weak non-interactive statistical zero-knowledge proof system,* where we ask only that for every polynomial $g(n)$, there exists a probabilistic polynomial-time simulator

[3] Actually, only non-interactive *perfect* and computational zero-knowledge proofs were defined in [6]. The definition we are using, previously given in [5, 12], is the natural non-interactive analogue of (interactive) statistical zero knowledge [29].

[4] Recall that a function is *negligible* if it is eventually less than $1/g(n)$ for any polynomial g.

S_g (whose running time may depend on g), such that the simulator deviation as defined above is at most $1/g(|x|)$. This is the natural analogue of a notion defined in the interactive setting for statistical zero knowledge [17] as well as concurrent zero knowledge [19].

The class of promise problems that possess non-interactive statistical zero-knowledge proof systems is denoted $\mathcal{NISZK}$, and we denote by *weak*-$\mathcal{NISZK}$ the class of promise problems that possess weak non-interactive statistical zero-knowledge proof systems. Note that by definition, $\mathcal{NISZK} \subset$ *weak*-$\mathcal{NISZK}$. De Santis *et. al.* [12] recently began a general investigation of the class $\mathcal{NISZK}$. They introduced a promise problem, called `Image Density`, and claimed that is complete for $\mathcal{NISZK}$ and that the latter class is closed under OR and complement. We were able to verify that some variants of `Image Density` are $\mathcal{NISZK}$-complete, and indeed the ideas used towards this goal are important to our work. However, they have retracted their claims that $\mathcal{NISZK}$ is closed under OR and complement [13].

In this paper, in addition to examining $\mathcal{NISZK}$ on its own, we also consider the relationship non-interactive statistical zero-knowledge proofs have with *interactive* statistical zero-knowledge proofs. In the context of interactive zero-knowledge proofs, another issue that arises in the zero-knowledge condition is the behavior of the verifier. The general definition of zero knowledge requires that the zero-knowledge requirement hold for any probabilistic polynomial-time verifier. A weaker requirement, called *honest-verifier zero knowledge,* requires the zero-knowledge condition to hold only if the verifier behaves honestly. However, it is known that these two conditions are equivalent for statistical zero knowledge, in the sense that every statistical zero-knowledge proof against the honest verifier can be transformed into one that is statistical zero knowledge against any verifier [25]. Thus, we write $\mathcal{SZK}$ for the class of promise problems possessing statistical zero-knowledge proofs (against any polynomial-time verifier or, equivalently, against just the honest verifier).

Note that in the case of non-interactive zero knowledge, the issue of honest verifiers does not arise since the verifier does not interact with the prover. Also, note that we can always transform a non-interactive zero-knowledge proof into an honest verifier zero-knowledge proof, since we could have the honest verifier supply a random string which can replace the common reference string required for non-interactive zero knowledge. That is, $\mathcal{NISZK} \subset \mathcal{SZK}$ (recalling the equivalence of $\mathcal{SZK}$ with honest-verifier $\mathcal{SZK}$).

1.2 Our Results

The primary tools we use in our investigation are promise problems that are complete for $\mathcal{SZK}$ or $\mathcal{NISZK}$. In [38], a promise problem called Statistical Difference (SD) was introduced and proved complete for $\mathcal{SZK}$, providing the first completeness result for $\mathcal{SZK}$. Recently, it was shown in [27] that another natural problem, called Entropy Difference (ED), is complete for $\mathcal{SZK}$ as well. In this work, we show that "one-sided" versions of these problems, which we call Statistical Difference from Uniform (SDU) and Entropy Approximation (EA), are complete for $\mathcal{NISZK}$. To define these problems more precisely, we first recall that that *statistical difference* between two random variables X and Y on a finite set D, denoted $\Delta(X\,,\,Y)$, is defined to be

$$\Delta(X\,,\,Y) \stackrel{\rm def}{=} \max_{S\subset D} |\Pr[X\in S] - \Pr[Y\in S]| = \frac{1}{2}\cdot\sum_{\alpha} |\Pr[X=\alpha] - \Pr[Y=\alpha]|.$$

All the promise problems we consider involve distributions which are encoded by circuits which sample from them. That is, if X is a circuit mapping $\{0,1\}^m$ to $\{0,1\}^n$, we identify X with the probability distribution induced on $\{0,1\}^n$ by feeding X the uniform distribution on $\{0,1\}^m$. Since circuits can be evaluated in time polynomial in their size, yet are rich enough to encode general (e.g., Turing machine) computations, they effectively capture the notion of an "efficiently sampleable distribution."

Definition 1.1. (Problems involving statistical difference): *The promise problem* Statistical Difference*, denoted* $\mathtt{SD} = (\mathtt{SD}_{\rm YES}, \mathtt{SD}_{\rm NO})$*, consists of*

$$\mathtt{SD}_{\rm YES} \stackrel{\rm def}{=} \{(X,Y) : \Delta(X\,,\,Y) < 1/3\}$$
$$\mathtt{SD}_{\rm NO} \stackrel{\rm def}{=} \{(X,Y) : \Delta(X\,,\,Y) > 2/3\}$$

where X and Y are distributions encoded as circuits which sample from them. Statistical Difference from Uniform*, denoted* $\mathtt{SDU} = (\mathtt{SDU}_{\rm YES}, \mathtt{SDU}_{\rm NO})$*, consists of*

$$\mathtt{SDU}_{\rm YES} \stackrel{\rm def}{=} \{X : \Delta(X\,,\,U) < 1/n\}$$
$$\mathtt{SDU}_{\rm NO} \stackrel{\rm def}{=} \{X : \Delta(X\,,\,U) > 1 - 1/n\}$$

where X is a distribution encoded as a circuit outputting n bits, and U is the uniform distribution on n bits.

For the two problems related to entropy, we recall that the (Shannon) entropy of a random variable X, denoted $\mathrm{H}(X)$, is defined as

$$\mathrm{H}(X) \stackrel{\rm def}{=} \sum_{\alpha} \Pr[X=\alpha]\cdot\log_2(1/\Pr[X=\alpha])$$

Definition 1.2. (Problems involving entropy): *The promise problem* Entropy Difference*, denoted* $\mathtt{ED} = (\mathtt{ED}_{\rm YES}, \mathtt{ED}_{\rm NO})$*, consists of*

$$\mathtt{ED}_{\rm YES} \stackrel{\rm def}{=} \{(X,Y) : \mathrm{H}(X) > \mathrm{H}(Y)+1\}$$
$$\mathtt{ED}_{\rm NO} \stackrel{\rm def}{=} \{(X,Y) : \mathrm{H}(Y) > \mathrm{H}(X)+1\}$$

Entropy Approximation*, denoted* $\mathtt{EA} = (\mathtt{EA}_{\rm YES}, \mathtt{EA}_{\rm NO})$*, consists of*

$$\mathtt{EA}_{\rm YES} \stackrel{\rm def}{=} \{(X,k) : \mathrm{H}(X) > k+1\}$$
$$\mathtt{EA}_{\rm NO} \stackrel{\rm def}{=} \{(X,k) : \mathrm{H}(X) < k-1\}$$

In these problems, k is a positive integer and X and Y are distributions encoded as circuits which sample from them.

Our first theorem, which is the starting point for our other results, is:

Theorem 1.3. (EA and SDU are $\mathcal{NISZK}$-complete) *The promise problems* EA *and* SDU *are complete for* $\mathcal{NISZK}$. *That is,* EA, SDU $\in \mathcal{NISZK}$ *and for every promise problem* $\Pi \in \mathcal{NISZK}$, *there is a polynomial-time Karp (many-one) reduction from* Π *to* EA *and another from* Π *to* SDU.

From the proof of this theorem, we also obtain a method for transforming weak non-interactive statistical zero knowledge proofs into standard ones.

Theorem 1.4. *weak-*$\mathcal{NISZK} = \mathcal{NISZK}$.

Armed with our complete problems, we then begin the work of comparing $\mathcal{SZK}$ and $\mathcal{NISZK}$. First we show that the non-triviality of $\mathcal{NISZK}$ is equivalant to the non-triviality of $\mathcal{SZK}$. This is shown by giving a Cook reduction from ED to EA.

Theorem 1.5. (non-triviality of $\mathcal{NISZK}$) $\mathcal{SZK} \neq \mathcal{BPP} \iff \mathcal{NISZK} \neq \mathcal{BPP}$.

In this theorem (and throughout the paper), $\mathcal{BPP}$ denotes the class of *promise problems* solvable in probabilistic polynomial time.

In fact, it turns out that the type of Cook reduction we use is a special one, and by examining it further, we are able to shed more light on the $\mathcal{SZK}$ vs. $\mathcal{NISZK}$ question. Specifically, we observe that the reduction we give from ED to EA is an $\mathcal{AC}^0$ *truth-table reduction.* That is, it is a nonadaptive Cook reduction in which the postprocessing is done in $\mathcal{AC}^0$. (Formal definitions are given in Section 5.2.) Further, we can prove that if $\mathcal{NISZK}$ is closed under complement, then $\mathcal{NISZK}$ is closed under $\mathcal{AC}^0$ truth-table reductions. Thus we deduce that $\mathcal{NISZK}$ being closed under complement implies that $\mathcal{NISZK} = \mathcal{SZK}$. In fact, we can show that closure under complement and a number of other natural conditions are equivalent to $\mathcal{SZK} = \mathcal{NISZK}$:

Theorem 1.6. (conditions for $\mathcal{SZK} = \mathcal{NISZK}$) *The following are equivalent:*

1. $\mathcal{SZK} = \mathcal{NISZK}$.
2. $\mathcal{NISZK}$ *is closed under complement.*
3. $\mathcal{NISZK}$ *is closed under* $\mathcal{NC}^1$ *truth-table reductions.*
4. ED *(resp.,* SD*) Karp-reduces to* EA *(resp.,* SDU*). ("general versions reduce to one-sided ones")*
5. EA *(resp.,* SDU*) Karp-reduces to its complement. ("one-sided versions reduce to their complements")*

Theorem 1.6 can be interpreted as saying that if $\mathcal{NISZK}$ has a relatively weak closure property (closure under complement), then the class is surprisingly rich (equals $\mathcal{SZK}$) and has a much stronger closure property (closure under $\mathcal{NC}^1$ truth-table reductions.) At first, it might seem implausible that a class like $\mathcal{NISZK}$ with such an assymetric definition would be closed under complement. But $\mathcal{SZK}$, which has a similarly assymetric definition, is known to be closed under complement [35]. In light of this, the closure of $\mathcal{NISZK}$ under complement would not be quite as unexpected, and Theorem 1.6 illustrates that proving it would have wider consequences.

The last two conditions in Theorem 1.6 show that these questions about non-interactive versus interactive statistical zero-knowledge proofs are actually equivalent to basic questions about relationships between natural computational problems whose definitions have no *a priori* relationship to zero-knowledge proofs.

The equality of $\mathcal{SZK}$ and $\mathcal{NISZK}$ has interesting consequences not just for $\mathcal{NISZK}$, but also for $\mathcal{SZK}$. Currently, the best known generic protocol for $\mathcal{SZK}$ (against cheating verifiers, making no computational assumptions) requires a polynomial number of rounds [35, 25].[5] For $\mathcal{NISZK}$, however, by [10], it is known that every problem in $\mathcal{NISZK}$ has a *constant round* statistical zero-knowledge proof system (against general, cheating verifiers) with inverse polynomial soundness error. Whether every problem in $\mathcal{SZK}$ has such a proof system is still an open question, which would be resolved in the positive if $\mathcal{SZK} = \mathcal{NISZK}$.

1.3 A Wider Perspective

The study of non-interactive *statistical* (rather than *computational*) zero-knowledge proofs may be of interest for two reasons. Firstly, *statistical* zero-knowledge proofs provide an almost absolute level of security, whereas *computational* zero-knowledge proofs only provide security relative to computational abilities (and typically under complexity theoretic assumptions). Secondly, by analogy from the study of zero-knowledge *interactive* proofs, we believe that techniques developed for the "cleaner" statistical model can be applied or augmented to yield results for computational zero-knowledge: The proof that one-way functions are necessary for $\mathcal{SZK}$ to be non-trivial [36] was later generalized to $\mathcal{CZK}$ [37]. More recently, the transformations of honest-verifier zero knowledge to general zero knowledge, presented in [8, 10, 9, 25], apply both to statistical and computational zero knowledge (whereas the original motivation was the study of statistical zero knowledge). It is our hope that the current study of $\mathcal{NISZK}$ will eventually lead to a better understanding of $\mathcal{NICZK}$, where there are still important open questions such as the minimal conditions under which $\mathcal{NP}$ has $\mathcal{NICZK}$ proofs.

2 Preliminaries

Recall that a *promise problem* Π is a pair $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ of disjoint sets of strings, corresponding to the following decision problem: Given a string $x \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$, decide whether it is in Π_{YES} (i.e. is a YES instance) or in Π_{NO} (i.e. is a NO instance). A string in $\Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ is said to *satisfy the promise*, and all

[5] Under the assumption that the Discrete Logarithm is hard, however, there is a constant round, cheating verifier $\mathcal{SZK}$ proof system with inverse polynomial soundness error for all of $\mathcal{SZK}$ [35, 4].

other strings are said to *violate the promise*. A function f is said to be a *Karp (or polynomial-time many-one) reduction* from a promise problem Π to a promise problem Γ if f is polynomial-time computable, $x \in \Pi_{\mathrm{YES}} \Rightarrow f(x) \in \Gamma_{\mathrm{YES}}$, and $x \in \Pi_{\mathrm{NO}} \Rightarrow f(x) \in \Gamma_{\mathrm{NO}}$. If such a reduction exists, we write $\Pi \leq_{\mathtt{Karp}} \Gamma$.

Recall that all the promise problems we are considering involve distributions which are encoded by circuits which sample from them. That is, if X is a circuit mapping $\{0,1\}^m$ to $\{0,1\}^n$, we identify X with the probability distribution induced on $\{0,1\}^n$ by feeding X the uniform distribution on $\{0,1\}^m$. The *support* of X is the set of strings in $\{0,1\}^n$ which have nonzero probability under X, i.e. $\{y \in \{0,1\}^n : \exists r \in \{0,1\}^m \text{ s.t. } X(r) = y\}$. For any distribution X on a set D, we write $\otimes^k X$ to denote the distribution on D^k consisting of k *independent* copies of X.

3 EA is in $\mathcal{NISZK}$

In this section, we show that EA has a non-interactive statistical zero-knowledge proof system.

Lemma 3.1. EA $\in \mathcal{NISZK}$. *Moreover, there is a non-interactive statistical zero-knowledge proof system for* EA *in which the completeness error, soundness error, and simulator deviation are all exponentially vanishing (specifically* 2^{-s}*, where* s *is the length of the input).*

The transformation given by the following lemma will be applied at the start of the proof system:

Lemma 3.2. *There is a polynomial-time computable function that takes an instance* (X, k) *of* EA *and a parameter* s *(in unary) and produces a distribution* Z *on* $\{0,1\}^\ell$ *(encoded by a circuit which samples from it) such that*

1. *If* $\mathrm{H}(X) > k + 1$*, then* Z *has statistical difference at most* 2^{-s} *from the uniform distribution on* $\{0,1\}^\ell$*, and*
2. *If* $\mathrm{H}(X) < k - 1$*, then the support of* Z *is at most a* 2^{-s} *fraction of* $\{0,1\}^\ell$*.*

The proof of Lemma 3.2 uses 2-universal hashing and the Leftover Hash Lemma [30] and is the most technically involved part of this work. However, due to space constraints, the construction and proof are deferred to the full version of the paper [26]. Lemma 3.2 essentially transforms an instance of `Entropy Approximation` into an instance of `Image Density`, the complete problem of [12]. Given this transformation, it is straightforward to give a noninteractive statistical zero-knowledge proof system for EA:

Non-interactive proof system for EA*, on input* (X, k)

1. Let Z be the distribution on $\{0,1\}^\ell$ obtained from (X, k) as in Lemma 3.2 taking s to be the total description length of (X, k) in bits. Let $\sigma \in \{0,1\}^\ell$ be the reference string.
2. P selects r uniformly among $\{r' : Z(r') = \sigma\}$ and sends r to V.

3. V accept if $Z(r) = \sigma$ and rejects otherwise.

It is immediate from Lemma 3.2 that the completeness error and soundness error of this proof system are 2^{-s}. For zero-knowledgeness, we consider the following probabilistic polynomial-time simulator:

Simulator for EA *proof system, on input* (X, k)

1. Let Z be obtained from (X, k) as in the proof system.
2. Select an input r to Z uniformly at random and let $\sigma = Z(r)$.
3. Output (σ, r).

It follows from Part 1 of Lemma 3.2 that this simulator has statistical difference at most 2^{-s} from the distribution of transcripts of (P, V). Thus, assuming Lemma 3.2, we have established Lemma 3.1. In fact, we need not require that s be the length of (X, k). Instead, s can be taken to be an arbitrary security parameter, and the completeness, soundness, and simulation error will be exponentially small in s, while the running time of the protocol only depends polynomially on s. We can use this to prove the following, which will be useful to us later.

Proposition 3.3. *If any promise problem Π reduces to* EA *by a Karp (i.e. many-one) reduction (even if it is length-reducing), then $\Pi \in \mathcal{NISZK}$.*

Proof. A noninteractive statistical zero-knowledge proof system for Π can be given as follows: On an instance x of Π, both parties compute the image (X, k) of x under the reduction $\Pi \leq_{\texttt{Karp}} \texttt{EA}$ and execute the proof system for EA on (X, k), except that we take s to be the length of x. Hence, the completeness and soundness errors and simulator deviation of this proof system are exponentially small in $|x|$ (rather than $|(X, k)|$ which could be shorter than x). □

4 EA and SDU are $\mathcal{NISZK}$-complete

In this section, we complete the proof of Theorem 1.3. First, we establish that $\texttt{SDU} \in \mathcal{NISZK}$ by showing:

Lemma 4.1. $\texttt{SDU} \leq_{\texttt{Karp}} \texttt{EA}$. *In particular,* $\texttt{SDU} \in \mathcal{NISZK}$.

Proof. Let X be an instance of SDU. We assume that $\log(n) > 5$, where n is the output length of the circuit X (otherwise, once can decide in probabilistic polynomial time whether X is a YES or NO instance of SDU by random sampling). Let U denote the uniform distribution on n bits. We claim the map $X \mapsto (X, n-3)$ is the reduction required by the lemma.

If $X \in \texttt{SDU}_{\text{YES}}$, then $\delta = \Delta(X\,,\,U) < 1/n$. Now we use the fact (see, e.g., [27]) that for any two random variables, Y and Z, ranging over domain D it holds that

$$|\mathrm{H}(Y) - \mathrm{H}(Z)| \;\leq\; (\log|D|) \cdot \Delta(Y\,,\,Z) \;+\; \mathrm{H}_2\,(\Delta(Y\,,\,Z))\,,$$

where $\mathrm{H}_2(\theta)$ denotes the entropy of a 0–1 random variable with mean θ. Applying this with $Y = U$ and $Z = X$, we have

$$n - \mathrm{H}(X) < n \cdot 1/n + \mathrm{H}_2(1/n) < 2.$$

Hence $(X, n-3) \in \mathtt{EA}_{\mathrm{YES}}$.

If $X \in \mathtt{SDU}_{\mathrm{NO}}$, then $\Delta(X\,, U) \geq 1 - 1/n$. By the definiton of statistical difference, this implies the existence of a set $S \subset \{0,1\}^n$ such that $\Pr[X \in S] - \Pr[U \in S] > 1 - 1/n$. This implies that

$$\Pr[X \in S] > 1 - 1/n \quad \text{and} \quad \Pr[U \in S] < 1/n.$$

Thus, $\mathrm{H}(X) \leq \Pr[X \in S] \cdot \log(|S|) + \Pr[X \notin S] \cdot n < 1 \cdot (n - \log n) + (1/n) \cdot n < n - 4$, and we have that $(X, n-3) \in \mathtt{EA}_{\mathrm{NO}}$.

The "in particular" part of Lemma 4.1 follows immediately from Proposition 3.3. □

Now, we establish both Theorem 1.3 and Theorem 1.4 by showing that all promise problems in weak-$\mathcal{NISZK}$ (and hence all promise problems in $\mathcal{NISZK}$) are reducible to SDU (and hence by the previous lemma to EA).

Lemma 4.2. *Every promise problem in weak-$\mathcal{NISZK}$ Karp-reduces to* SDU.

Proof. Let Π be any promise problem in *weak-*$\mathcal{NISZK}$. As *weak-*$\mathcal{NISZK}$ is preserved under parallel repetition, we may assume that Π has a *weak-*$\mathcal{NISZK}$ proof system (P, V) with completeness and soundness errors at most 2^{-n} on inputs of length n. Let $r(n) = \mathrm{poly}(n)$ be the length of the random reference string in (P, V), and let S be a randomized polynomial-time simulator S such that the statistical difference between the output distribution of S and the distribution of true transcripts of P is at most $1/(3r(n))$. (Such an S is guaranteed by the *weak-*$\mathcal{NISZK}$ property.) Let U denote the uniform distribution on $r(n)$ bits.

Let x be an instance of Π. Define M_x to be a circuit which does the following on input s:
$M_x(s)$: Simulate $S(x)$ with randomness s to obtain a transcript (σ, p). If $V(x, \sigma, p)$ accepts, then output σ, else output $0^{r(n)}$.

We claim that the map $x \mapsto M_x$ is the reduction required by the lemma. Suppose $x \in \Pi_{\mathrm{YES}}$. In this case, we know that the random reference string σ in the output of S has statistical difference less than $1/3r(n)$ from U. In addition, since the completeness error of protocol P is at most 2^{-n}, $S(x)$ can output rejecting transcripts with probability at most $1/(3r(n)) + 2^{-n} \leq 2/(3r(n))$. Hence, $\Delta(M_x\,, U) < 2/(3r(n)) + 1/(3r(n)) \leq 1/r(n)$, and $M_x \in \mathtt{SDU}_{\mathrm{YES}}$.

Suppose $x \in \Pi_{\mathrm{NO}}$. Since the soundness error of protocol P is bounded by 2^{-n}, for at most a 2^{-n} fraction of reference strings σ does there exist an accepting transcript (σ, p). Since M_x only outputs reference strings corresponding to accepting transcripts or $0^{r(n)}$, $\Delta(M_x\,, U) \geq 1 - (2^{-n} + 2^{-r(n)}) > 1 - 1/r(n)$. Thus, $M_x \in \mathtt{SDU}_{\mathrm{NO}}$. □

Clearly, Lemmas 3.1, 4.1, and 4.2 combine to prove Theorem 1.3. Lemmas 4.2 and 4.1 show that any promise problem Π in weak-$\mathcal{NISZK}$ reduces to EA; by Proposition 3.3, this implies that $\Pi \in \mathcal{NISZK}$ and establishes Theorem 1.4.

5 Comparing $\mathcal{NISZK}$ and $\mathcal{SZK}$

Armed with $\mathcal{NISZK}$-complete promise problems so closely related to problems known to be complete for $\mathcal{SZK}$, we can quickly begin relating the two classes.

5.1 Nontriviality of $\mathcal{NISZK}$

First, we establish Theorem 1.5 by giving a Cook reduction from `Entropy Difference` (`ED`), complete for $\mathcal{SZK}$, to `Entropy Approximation` (`EA`), complete for $\mathcal{NISZK}$.

Lemma 5.1. *Suppose (X, Y) is an instance of* `ED`*. Let $X' = \otimes^4 X$ (resp., $Y' = \otimes^4 Y$) consist of 4 independent copies of X (resp., Y), and let n denote the maximum of the output sizes of X' and Y'. Then,*

$$(X, Y) \in \mathtt{ED}_{\mathrm{YES}} \Longrightarrow \bigvee_{k=1}^{n} [((X', k) \in \mathtt{EA}_{\mathrm{YES}}) \wedge ((Y', k) \in \mathtt{EA}_{\mathrm{NO}})]$$

$$(X, Y) \in \mathtt{ED}_{\mathrm{NO}} \Longrightarrow \bigwedge_{k=1}^{n} [((X', k) \in \mathtt{EA}_{\mathrm{NO}}) \vee ((Y', k) \in \mathtt{EA}_{\mathrm{YES}})]$$

Proof. Suppose $(X, Y) \in \mathtt{ED}_{\mathrm{YES}}$, so that $H(X') > H(Y') + 4$. Let $k = \lfloor H(X') \rfloor - 2$. Then $H(X') > k+1$. On the other hand, $k+3 > H(X') > H(Y')+4$, and hence $H(Y') < k - 1$. Suppose instead $(X, Y) \in \mathtt{ED}_{\mathrm{NO}}$, so that $H(Y') > H(X') + 4$. Then for all $k > \lceil H(X') \rceil + 1$, we have $H(X') < k-1$. So, for all $k \leq \lceil H(X') \rceil + 1$, we have $k + 1 < H(X') + 3 < H(Y')$.

From this reduction, we conclude that $\mathcal{SZK} \neq \mathcal{BPP} \iff \mathcal{NISZK} \neq \mathcal{BPP}$, which is Theorem 1.5. Again, by $\mathcal{BPP}$ we mean the class of *promise problems* solvable in probabilistic polynomial time.

Proof of Theorem 1.5. By definition, $\mathcal{NISZK} \subset \mathcal{SZK}$ (recall that $\mathcal{SZK}$ equals honest-verifier $\mathcal{SZK}$ [25]). Hence if $\mathcal{SZK} = \mathcal{BPP}$, then $\mathcal{NISZK} = \mathcal{BPP}$.

Now suppose $\mathcal{NISZK} = \mathcal{BPP}$, so in particular there is a probabilistic polynomial-time machine M which decides `EA` (with exponentially small error probability). To show $\mathcal{SZK} = \mathcal{BPP}$, it suffices to show that $\mathtt{ED} \in \mathcal{BPP}$ since `ED` is $\mathcal{SZK}$-complete. We now describe how to decide instances of `ED`: Let (X, Y) be an instance of `ED`. Letting X' and Y' be as stated in Lemma 5.1, we run $M(X', k)$ and $M(Y', k)$ for all $k \in [1, n]$. If for some k, we see that $M(X', k) = 1$ and $M(Y', k) = 0$, we output 1. Otherwise, we output 0. By Lemma 5.1, this is a correct $\mathcal{BPP}$ algorithm for deciding `ED`. □.

5.2 Conditions under which $\mathcal{NISZK} = \mathcal{SZK}$

The reduction given by Lemma 5.1 is a very special type of Cook reduction, which we call an $\mathcal{AC}^0$ truth-table reduction. In this section, we use the special properties of this reduction to show that if $\mathcal{NISZK}$ is closed under complement, then in fact $\mathcal{NISZK} = \mathcal{SZK}$. We now precisely define the types of reductions we are using, taking care how they are defined for promise problems.

Definition 5.2. (truth-table reduction [32]): *We say a promise problem Π* truth-table reduces *to a promise problem Γ, written $\Pi \leq_{\mathtt{tt}} \Gamma$, if there exists a (deterministic) polynomial-time computable function f, which on input x produces a tuple $(x_1, x_2, \ldots, x_k)$ and a circuit C, such that*

1. *If $x \in \Pi_{\mathrm{YES}}$ then for all valid settings of $b_1, b_2, \ldots, b_k$, $C(b_1, b_2, \ldots, b_k) = 1$, and*
2. *If $x \in \Pi_{\mathrm{NO}}$ then for all valid settings of $b_1, b_2, \ldots, b_k$, $C(b_1, b_2, \ldots, b_k) = 0$.*

where a setting for b_i is considered valid when $b_i = 1$ if $x_i \in \Gamma_{\mathrm{YES}}$ and $b_i = 0$ if $x_i \in \Gamma_{\mathrm{NO}}$ (and b_i is unrestricted when x_i violates the promise).

In other words, a truth-table reduction for promise problems is a non-adaptive Cook reduction which is allowed to make queries which violate the promise, but must be able to tolerate both yes and no answers in response to queries that violate the promise. We further consider the case where we restrict the complexity of computing the output of the reduction from the queries:

Definition 5.3. ($\mathcal{AC}^0$ and $\mathcal{NC}^1$ truth-table reductions): *A truth-table reduction f between promise problems is an $\mathcal{AC}^0$* (resp., $\mathcal{NC}^1$) truth-table reduction *if the circuit C produced by the reduction on input x has depth bounded by a constant c_f independent of x (resp., has depth bounded by $c_f \log |x|$). If there is an $\mathcal{AC}^0$ (resp., $\mathcal{NC}^1$) truth-table reduction from Π to Γ, we write $\Pi \leq_{\mathcal{AC}^0-\mathtt{tt}} \Gamma$ (resp., $\Pi \leq_{\mathcal{NC}^1-\mathtt{tt}} \Gamma$).*

With this definition, we observe that Lemma 5.1 in fact gives an $\mathcal{AC}^0$ truth-table reduction, since the formula given in the lemma can be expressed as an $\mathcal{AC}^0$ circuit, and the statement of the lemma shows that the reduction has the robustness properties against promise violations that are required in Definition 5.3. Thus, we have:

Proposition 5.4. $\mathtt{ED} \leq_{\mathcal{AC}^0-\mathtt{tt}} \mathtt{EA}$.

We say that a class $\mathcal{C}$ of promise problems is closed under a class of reductions $\leq_*$ if $\Pi \leq_* \Gamma$ and $\Gamma \in \mathcal{C}$ implies that $\Pi \in \mathcal{C}$. By the above, if $\mathcal{NISZK}$ is closed under $\mathcal{AC}^0$ truth-table reductions, then $\mathtt{ED} \in \mathcal{NISZK}$ and hence $\mathcal{NISZK} = \mathcal{SZK}$. Thus, we would like to capture the minimal conditions necessary for a promise class to be closed under $\mathcal{AC}^0$ truth-table reductions. Here, care must be taken to because of the possibility of promise violations. Keeping this in mind, we define the following operator on promise problems to capture the notion of an unbounded fan-in AND gate for promise problems:

Definition 5.5. *(unbounded AND): For any promise problem Π, we define $\mathtt{AND}(\Pi)$ to be the promise problem:*

$$\mathtt{AND}_{\mathrm{YES}}(\Pi) \stackrel{\mathrm{def}}{=} \{(x_1, x_2, \ldots, x_k) : k \geq 0, \forall i \in [1, k] x_i \in \Pi_{\mathrm{YES}}\}$$
$$\mathtt{AND}_{\mathrm{NO}}(\Pi) \stackrel{\mathrm{def}}{=} \{(x_1, x_2, \ldots, x_k) : k \geq 0, \exists i \in [1, k] x_i \in \Pi_{\mathrm{NO}}\}$$

We say a class of promise problems $\mathcal{C}$ is closed under unbounded AND *if $\Pi \in \mathcal{C}$ implies that $\mathtt{AND}(\Pi) \in \mathcal{C}$.*

We have defined AND so that it has the weakest promise condition possible to remain well-defined. In particular, we see that $\texttt{AND}_{\mathrm{NO}}(\Pi)$ is defined to include x_i's that violate Π's promise, as long as just *one* of them is in Π_{NO}. $\Pi \in \mathcal{C}$, $\texttt{AND}(\Pi) \in \mathcal{C}$. We also need a way of combining two promise problems:

Definition 5.6. *(disjoint union): For any pair of promise problems Π and Γ, we define the* disjoint union *of Π and Γ to be the promise problem* $\texttt{DisjUn}(\Pi, \Gamma)$ *defined as follows:*

$$\texttt{DisjUn}_{\mathrm{YES}}(\Pi, \Gamma) \stackrel{\mathrm{def}}{=} \{0\} \times \Pi_{\mathrm{YES}} \cup \{1\} \times \Gamma_{\mathrm{YES}}$$
$$\texttt{DisjUn}_{\mathrm{NO}}(\Pi, \Gamma) \stackrel{\mathrm{def}}{=} \{0\} \times \Pi_{\mathrm{NO}} \cup \{1\} \times \Gamma_{\mathrm{NO}}$$

We say a class of promise problems $\mathcal{C}$ is closed under disjoint union *if $\Pi, \Gamma \in \mathcal{C}$ implies that* $\texttt{DisjUn}(\Pi, \Gamma) \in \mathcal{C}$.

With these definitions, we can give the following lemma which gives some conditions sufficient to give closure under AC^0 truth-table reductions.

Lemma 5.7. *A promise class $\mathcal{C}$ is closed under $\mathcal{AC}^0$ truth-table reductions if the following conditions hold:*

1. *$\mathcal{C}$ is closed under Karp (i.e., many-one) reductions.*
2. *$\mathcal{C}$ is closed under unbounded AND.*
3. *$\mathcal{C}$ is closed under disjoint union.*
4. *$\mathcal{C}$ is closed under complementation.*

Lemma 5.7 can be proven by a straightforward induction on the depth of the circuits. Details are given in the full version of the paper [26]. Which of the conditions of Lemma 5.7 does $\mathcal{NISZK}$ satisfy? We argue that Conditions 1, 2, and 3 are satisfied by $\mathcal{NISZK}$: Closure under Karp reductions and disjoint union follows readily from Proposition 3.3 and the completeness of EA. For closure under unbounded AND, note that to give an $\mathcal{NISZK}$ proof for the AND of many statements, one can give individual $\mathcal{NISZK}$ proofs for each of the statements in parallel. The only technical difficulty is that the lengths of the statements are not guaranteed to be polynomially related, but this can be dealt with as in the proof of Proposition 3.3 or by noting that instances of EA can be trivially padded. Thus, we have the following lemmas (whose full proofs are given in the full version of this paper [26]):

Lemma 5.8. *$\mathcal{NISZK}$ is closed under Karp reductions.*

Lemma 5.9. *$\mathcal{NISZK}$ is closed under unbounded AND.*

Lemma 5.10. *$\mathcal{NISZK}$ is closed under disjoint union.*

Combining everything, we can give a condition under which $\mathcal{SZK} = \mathcal{NISZK}$.

Proposition 5.11. *If $\mathcal{NISZK}$ is closed under complementation, then $\mathcal{SZK} = \mathcal{NISZK}$.*

Proof. Suppose $\mathcal{NISZK}$ is closed under complementation. Combining this with Lemmas 5.7, 5.8, 5.9, and 5.10, it follows that $\mathcal{NISZK}$ is closed under $\mathcal{AC}^0$ truth-table reductions. Applying Proposition 5.4 ($\texttt{ED} \leq_{\mathcal{AC}^0-\texttt{tt}} \texttt{EA}$) and Lemma 3.1 ($\texttt{EA} \in \mathcal{NISZK}$), we conclude that $\texttt{ED} \in \mathcal{NISZK}$. Since ED is complete for $\mathcal{SZK}$ [27] and $\mathcal{NISZK}$ is closed under Karp reductions (Lemma 5.8), we have $\mathcal{SZK} \subset \mathcal{NISZK}$. As $\mathcal{NISZK} \subset \mathcal{SZK}$ is true from the definition of $\mathcal{NISZK}$, we conclude that $\mathcal{NISZK} = \mathcal{SZK}$. □

Finally, we deduce Theorem 1.6, which gives a number of conditions equivalent to $\mathcal{NISZK} = \mathcal{SZK}$.

Proof of Theorem 1.6:
1 ⇒ 3. This follows from the result of [39] that $\mathcal{SZK}$ is closed under $\mathcal{NC}^1$ truth-table reductions.
3 ⇒ 2 ⇒ 1. The first is trivial and the second is Proposition 5.11.
1 ⇔ 4. This follows from Theorem 1.3 (which asserts that that EA and SDU are complete for $\mathcal{NISZK}$), the fact that ED and SD are complete for $\mathcal{SZK}$ [38, 27], and Lemma 5.8 (that $\mathcal{NISZK}$ is closed under Karp reductions).
2 ⇔ 5. This follows from Theorem 1.3 (that EA and SDU are complete for $\mathcal{NISZK}$) and Lemma 5.8 (that $\mathcal{NISZK}$ is closed under Karp reductions).

Acknowledgments

The first author was supported by DARPA grant DABT63-96-C-0018. The second and third authors were supported by DOD/NDSEG Graduate Fellowships and partially by DARPA grant DABT63-96-C-0018. We thank the CRYPTO ‘99 program committee and reviewers for helpful comments on the presentation.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the Thirty Third Annual Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[2] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs. In *Proceedings of the Thirty Third Annual Symposium on Foundations of Computer Science*, pages 2–13, 1992.

[3] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In G. Brassard, editor, *Advances in Cryptology—CRYPTO ’89*, volume 435 of *Lecture Notes in Computer Science*, pages 194–211. Springer-Verlag, 1990, 20–24 August 1989.

[4] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 482–493, 1990.

[5] Mihir Bellare and Phillip Rogaway. Non-interactive perfect zero-knowledge. Unpublished manuscript, June 1990.

[6] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, December 1991.

[7] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 103–112, Chicago, Illinois, 2–4 May 1988.

[8] Ivan Damgård. Interactive hashing can simplify zero-knowledge protocol design. In *Proceedings of Crypto '95, Lecture Notes in Computer Science*, volume 403, pages 100–109. Springer-Verlag, 1994.

[9] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs. dishonest verifier in public coin zero-knowledge proofs. In *Proceedings of Crypto '95, Lecture Notes in Computer Science*, volume 403. Springer-Verlag, 1995.

[10] Ivan Damgård, Oded Goldreich, and Avi Wigderson. Hashing functions can simplify zero-knowledge protocol design (too). Technical Report RS-94–39, BRICS, November 1994. See Part 1 of [9].

[11] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Randomness-efficient non-interactive zero-knowledge (extended abstract). In Pierpaolo Degano, Robert Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium*, volume 1256 of *Lecture Notes in Computer Science*, pages 716–726, Bologna, Italy, 7–11 July 1997. Springer-Verlag.

[12] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. Image Density is complete for non-interactive-SZK. In *Automata, Languages and Programming, 25th International Colloquium*, Lecture Notes in Computer Science, pages 784–795, Aalborg, Denmark, 13–17 July 1998. Springer-Verlag. See [13].

[13] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. Image Density is complete for non-interactive-SZK, May 1999. Preliminary draft of full version.

[14] Alfredo De Santis, Giovanni Di Crescenzo, and Guiseppe Persiano. The knowledge complexity of quadratic residuosity languages. *Theoretical Computer Science*, 132(1–2):291–317, 26 September 1994.

[15] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 52–72. Springer-Verlag, 1988, 16–20 August 1987.

[16] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge with preprocessing. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 269–282. Springer-Verlag, 1990, 21–25 August 1988.

[17] Giovanni Di Crescenzo, Tatsuaki Okamoto, and Moti Yung. Keeping the SZK-verifier honest unconditionally. In *Advances in Cryptology—CRYPTO '97*, pages 31–45, 1997.

[18] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, Louisiana, 6–8 May 1991.

[19] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 409–418, 1998.

[20] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science*, volume I, pages 308–317, St. Louis, Missouri, 22–24 October 1990. IEEE.
[21] Lance Fortnow. The complexity of perfect zero-knowledge. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.
[22] Oded Goldreich and Eyal Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97–116, 1993.
[23] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the Association for Computing Machinery*, 38(1):691–729, 1991.
[24] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, Winter 1994.
[25] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 399–408, 1998.
[26] Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made non-interactive? or On the relationship of SZK and NISZK. Submitted to *ECCC*, May 1999. `http://www.eccc.uni-trier.de/eccc/`.
[27] Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 54–73, Atlanta, GA, May 1999. IEEE Computer Society Press.
[28] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
[29] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.
[30] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, 15–17 May 1989.
[31] Joe Kilian and Erez Petrank. An efficient noninteractive zero-knowledge proof system for NP with general assumptions. *Journal of Cryptology*, 11(1):1–27, Winter 1998.
[32] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–123, December 1975.
[33] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proofs. In *Proceedings of the Thirty First Annual Symposium on Foundations of Computer Science*, pages 1–10, 1990.
[34] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, Maryland, 14–16 May 1990.
[35] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. In *Proceedings of the Twenty Eighth Annual ACM Symposium on the Theory of Computing*, 1996. See also preprint of full version, Oct. 1997.
[36] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the Thirty Second Annual Symposium on Foundations of Computer Science*, pages 133–138, 1991.
[37] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the Second Israel Symposium on Theory of Computing and Systems*, 1993.

[38] Amit Sahai and Salil Vadhan. A complete promise problem for statistical zero-knowledge. In *Proceedings of the Thirty Eighth Annual Symposium on Foundations of Computer Science*, pages 448–457, 1997.
[39] Amit Sahai and Salil Vadhan. Manipulating statistical difference. In Panos Pardalos, Sanguthevar Rajasekaran, and José Rolim, editors, *Randomization Methods in Algorithm Design (DIMACS Workshop, December 1997)*, volume 43 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 251–270. American Mathematical Society, 1999.
[40] Adi Shamir. IP=PSPACE. In *Proceedings of the Thirty First Annual Symposium on Foundations of Computer Science*, pages 11–15, 1990.
[41] Andrew C. Yao. Theory and application of trapdoor functions. In *Proceedings of the Twenty Third Annual Symposium on Foundations of Computer Science*, pages 80–91, 1982.

A Definitions

Following [22], we extend the standard definition of interactive proof systems to promise problems $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$. That is, we require the completeness condition to hold for YES instances (i.e., $x \in \Pi_{\mathrm{YES}}$), require the soundness condition to hold for NO instances (i.e., $x \in \Pi_{\mathrm{NO}}$), and do not require anything for inputs which violate the promise (i.e., $x \notin \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$).

We are mainly interested in such proof systems which are statistical zero knowledge:

Definition A.1. (Statistical Zero Knowledge – $\mathcal{SZK}$): *Let (P, V) be an interactive proof system for a promise problem $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$.*

- *We denote by $\langle P, V\rangle(x)$ the* view *of the verifier V while interacting with P on common input x; this consists of the common input, V's internal coin tosses, and all messages it has received.*
- *(P, V) is said to be (general)* statistical zero knowledge *if, for every probabilistic polynomial-time V^*, there exists a probabilistic polynomial-time machine* (called a simulator)*, S, and a negligible function $\mu : \mathbb{N} \mapsto [0, 1]$* (called the simulator deviation) *so that for every $x \in \Pi_{\mathrm{YES}}$ the statistical difference between $S(x)$ and $\langle P, V^*\rangle(x)$ is at most $\mu(|x|)$.*
- *$\mathcal{SZK}$ denotes the class of promise problems having statistical zero-knowledge interactive proof systems.*

Honest-verifier statistical zero-knowledge proof systems are such where the zero-knowledge requirement is only required to hold for the prescribed/honest verifier V, rather than for every polynomial-time computable V^*. Every honest-verifier statistical zero-knowledge proof system can be transformed into a general statistical zero-knowledge proof system (actually meeting an even stronger zero-knowledge requirement) [25].