Electronic Colloquium on Computational Complexity, Report No. 139 (2006)

Zero Knowledge and Soundness are Symmetric*

Shien Jin Ong Salil Vadhan

Division of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts, USA.
E-mail: {shienjin,salil}@eecs.harvard.edu

November 14, 2006

Abstract

We give a complexity-theoretic characterization of the class of problems in **NP** having zero-knowledge argument systems that is symmetric in its treatment of the zero knowledge and the soundness conditions. From this, we deduce that the class of problems in $\mathbf{NP} \cap \mathbf{coNP}$ having zero-knowledge arguments is closed under complement. Furthermore, we show that a problem in **NP** has a *statistical* zero-knowledge argument system if and only if its complement has a computational zero-knowledge *proof* system. What is novel about these results is that they are *unconditional*, i.e. do not rely on unproven complexity assumptions such as the existence of one-way functions.

Our characterization of zero-knowledge arguments also enables us to prove a variety of other unconditional results about the class of problems in **NP** having zero-knowledge arguments, such as equivalences between honest-verifier and malicious-verifier zero knowledge, private coins and public coins, inefficient provers and efficient provers, and non-black-box simulation and black-box simulation. Previously, such results were only known unconditionally for zero-knowledge *proof systems*, or under the assumption that one-way functions exist for zero-knowledge argument systems.

Keywords: zero-knowledge argument systems, statistical zero knowledge, complexity classes, closure under complement, distributional one-way functions.

*Both the authors were supported by NSF grant CNS-0430336 and ONR grant N00014-04-1-0478.

ISSN 1433-8092

1 Introduction

Zero-knowledge protocols are interactive protocols whereby one party, the *prover*, convinces another party, the *verifier*, that some assertion is true with the remarkable property that the verifier "learns nothing" other than the fact that the assertion being proven is true. Since their introduction by Goldwasser, Micali, and Rackoff [GMR89], zero-knowledge protocols have played a central role in the design and study of cryptographic protocols.

Zero-knowledge protocols come in several flavors, depending on how one formulates the two security conditions: (1) the zero-knowledge condition, which says that the verifier "learns nothing" other than the fact the assertion being proven is true, and (2) the soundness conditions, which says that the prover cannot convince the verifier of a false assertion. In *statistical zero knowledge*, the zero-knowledge condition holds regardless of the computational resources the verifier invests into trying to learn something from the interaction. In *computational zero knowledge*, we only require that a probabilistic polynomial-time verifier learn nothing from the interaction.[1] Similarly, for soundness, we have *statistical soundness*, also known as *proof systems*, where even a computationally unbounded prover cannot convince the verifier of a false statement (except with negligible probability), and *computational soundness*, also known as *argument systems* [BCC88], where we only require that a polynomial-time prover cannot convince the verifier of a false statement. Using a prefix of **S** or **C** to indicate whether the zero knowledge is statistical or computational and a suffix of **P** or **A** to indicate whether we have a proof system or argument system, we obtain four complexity classes corresponding to the different types of zero-knowledge protocols: **SZKP**, **CZKP**, **SZKA**, **CZKA**. More precisely, these are the classes of *decision problems* Π having the correponding type of zero-knowledge protocol. In such a protocol, the prover and verifier are given as common input an instance x of Π, and the prover is trying convince the verifier that x is a YES instance of Π.

These two security conditions seem to be of very different flavors; zero knowledge is a 'secrecy' condition, whereas soundness is more of an 'unforgeability' condition. However, in a remarkable paper, Okamoto [Oka00] showed that they are actually symmetric in the case of statistical security.

Theorem 1.1 ([Oka00])**.** [2] *The class* **SZKP** *of problems having statistical zero-knowledge proofs is closed under complement. That is,* $\Pi \in \mathbf{SZKP}$ *if and only if* $\overline{\Pi} \in \mathbf{SZKP}$*.*

In a zero-knowledge protocol for proving that a string x is a YES instance of a problem Π, zero knowledge is required only when x is indeed a YES instance (i.e. the statement being proven is true) and soundness is required only when x is a NO instance (i.e. the statement is false). Thus, by showing that **SZKP** is closed under complement, Okamoto established a symmetry between zero knowledge and soundness, in the case that both security conditions are statistical.

It is natural to ask whether an analogous theorem holds when the security conditions are *computational*, i.e. for computational zero-knowledge arguments. If we assume the existence of one-way functions, then the answer is yes. Indeed, classic results of [GMW91, Nao91, HILL99] show that every problem in **NP** has a computational zero-knowledge argument system if one-way functions exist, and in fact it is known that either one (but not both) of the security conditions can be made statistical (cf., [NOV06]). (Here, and throughout the paper, we usually restrict

[1] More precisely, in statistical zero knowledge, we require that the verifier's view of the interaction can be efficiently simulated up to negligible statistical distance, whereas in computational zero knowledge, we only require that the simulation be computationally indistinguishable from the verifier's view.

[2] Okamoto's result was actually for the class of languages having *honest-verifier* statistical zero-knowledge proofs, but in [GSV98] it was shown this is the same as the class of languages having general statistical zero-knowledge proofs.

attention to problems in **NP**, because argument systems are mainly of interest when the prover can be implemented in polynomial time given a witness of membership, which only makes sense for problems in **NP**.[3]) Thus, if one-way functions exist, symmetry between computational zero knowledge and computational soundness holds for problems in $\mathbf{NP} \cap \mathbf{coNP}$, by virtue that all problems in $\mathbf{NP} \cap \mathbf{coNP}$ and their complements have computational zero-knowledge arguments (assuming one-way functions).

In this paper, we establish an *unconditional* symmetry between computational zero knowledge and computational soundness:

Theorem 1.2 (Symmetry Theorem)**.**

1. *(***CZKA** *vs.* **co-CZKA***) A problem* $\Pi \in \mathbf{NP} \cap \mathbf{coNP}$ *has a computational zero-knowledge argument system if and only if* $\overline{\Pi}$ *has a computational zero-knowledge argument system.*

2. *(***SZKA** *vs.* **CZKP***) A problem* $\Pi \in \mathbf{NP}$ *has a statistical zero-knowledge argument system if and only if* $\overline{\Pi}$ *has a computational zero-knowledge proof system.*

Note how the quality of the zero-knowledge condition for Π translates to the quality of the soundness condition for $\overline{\Pi}$ and vice-versa.

1.1 The SZKP–OWF Triplet Characterizations

The Symmetry Theorem is obtained by new characterizations of the classes of problems having various types of zero-knowledge protocols, and moreover these characterizations treat zero knowledge and soundness symmetrically. These characterizations are a generalization of the "SZK/OWF Condition" of [Vad04], which says that any problem having a computational zero-knowledge proof can be described as a problem having a statistical zero-knowledge proof plus a set of YES instances from which we can construct a one-way function. To characterize argument systems, we will also allow some additional NO instances from which we can construct a one-way function.

To formalize this, we will need the notion of a *promise problem*, which is simply decision problem with some inputs excluded. More precisely, a promise problem Π consists of two disjoint sets of strings (Π_Y, Π_N), corresponding to YES and NO instances respectively. All of the complexity classes we are using (e.g. $\mathbf{SZKP}, \mathbf{CZKP}, \mathbf{SZKA}, \mathbf{CZKA}$) generalize to promise problems in the natural way; completeness and zero knowledge are required for YES instances and soundness is required for NO instances.

Definition 1.3 (SZKP–OWF Triplets)**.** Let $\Pi = (\Pi_Y, \Pi_N)$ be a promise problem. We say that (Π, I, J) is a SZKP–OWF Triplet if the following three conditions hold:

1. The promise problem $(\Pi_Y \setminus I, \Pi_N \setminus J)$ is in **SZKP**.

2. There exists a polynomial-time computable function $f_x(y) \stackrel{\text{def}}{=} f(x,y)$ such that f_x is one-way for every $x \in I$. That is, there exists a polynomial $p(\cdot)$ such that for every nonuniform polynomial-time algorithm A, and every $x \in I$,

$$\Pr\left[A(f_x(U_{p(|x|)}) \in f_x^{-1}(f_x(U_{p(|x|)}))\right] \leq \mathrm{neg}(|x|).$$

[3]Actually polynomial-time provers also make sense for problems in **MA**, which is a variant of **NP** where the verification of witnesses is probabilistic. All of our results easily extend to **MA**, but we state them for **NP** for simplicity.

3. There exists a polynomial-time computable function $g_x(y) \stackrel{\text{def}}{=} g(x,y)$ such that g_x is one-way for every $x \in J$. That is, there exists a polynomial $q(\cdot)$ such that for every nonuniform polynomial-time algorithm A, and every $x \in J$,

$$\Pr\left[A(g_x(U_{q(|x|)}) \in g_x^{-1}(g_x(U_{q(|x|)}))\right] \leq \text{neg}(|x|).$$

We use this to characterize the classes of problems having zero-knowledge protocols as follows.

Theorem 1.4 (SZKP–OWF TRIPLET Characterization)**.**

1. *(***SZKP** [trivial]*) A problem* $\Pi \in$ **IP** *has a statistical zero-knowledge proof system if and only if* $(\Pi, \emptyset, \emptyset)$ *is a* SZKP–OWF TRIPLET*.*
2. *(***CZKP** [Vad04]*) A problem* $\Pi \in$ **IP** *has a computational zero-knowledge proof system if and only if there exists a set* $I \subseteq \Pi_{\text{Y}}$ *such that* $(\Pi, I, \emptyset)$ *is a* SZKP–OWF TRIPLET*.*
3. *(***CZKA** [new]*) A problem* $\Pi \in$ **NP** *has a computational zero-knowledge argument system if and only if there exist sets* $I \subseteq \Pi_{\text{Y}}$*,* $J \subseteq \Pi_{\text{Y}}$ *such that* (Π, I, J) *is a* SZKP–OWF TRIPLET*.*
4. *(***SZKA** [new]*) A problem* $\Pi \in$ **NP** *has a statistical zero-knowledge argument system if and only if there exists a set* $J \subseteq \Pi_{\text{N}}$ *such that* $(\Pi, \emptyset, J)$ *is a* SZKP–OWF TRIPLET*.*

Notice the symmetric roles of the "OWF instances" I and J, with each being empty corresponding to a statistical security condition, in zero knowledge and completeness, respectively. Theorem 1.2 follows immediately from Theorem 1.4 together with the fact that **SZKP** is closed under complement (Theorem 1.1).

The advantage of the SZKP–OWF TRIPLET characterizations is that they reduce the study of the various forms of zero-knowledge protocols to the study of **SZKP** together with the study of the consequences of one-way functions, both of which are by now quite well-developed. Indeed, we also use these characterizations to prove many other unconditional theorems about the classes of problems in **NP** possessing zero-knowledge arguments, such as equivalences between honest-verifier and malicious-verifier zero knowledge, private coins and public coins, inefficient provers and efficient provers, and non-black-box simulation and black-box simulation. Previously, such results were only known unconditionally for zero-knowledge *proof systems* [Oka00, GSV98, Vad04, NV06], or were known under the assumption that one-way functions exist for zero-knowledge argument systems [GMW91, Nao91, HILL99, NOV06].

While our characterizations of **CZKA** and **SZKA** (Items 3 and 4) are similar in spirit to the **CZKP** characterization of [Vad04] (Item 2), both directions of the implications require new ingredients that were not present in [Vad04].

In the forward direction, going from **CZKA** or **SZKA** to an SZKP–OWF TRIPLET, we combine [Vad04] with the work of Ostrovsky [Ost91] to construct the one-way function on instances in J. Ostrovsky showed that if a *hard-on-average* problem has a statistical zero-knowledge argument system, then (standard) one-way functions exist.[4] (This was later generalized to computational zero knowledge in [OW93].) We use the same construction, but with a slightly different analysis. In Ostrovsky's work the hardness of inverting the one-way function is derived from the assumed (average-case) hardness of the problem having the zero-knowledge protocol, and it is shown to be hard to invert on YES instances. In our proof, the hardness of inverting the one-way function is

[4]Ostrovky's theorem is only stated in terms of only refers to statistical zero-knowledge proofs, but it immediately extends to arguments.

instead derived from a gap between between statistical soundness and computational soundness, and it is analyzed on NO instances.

In the reverse direction, going from an SZKP–OWF TRIPLET to **CZKA** or **SZKA**, there were more fundamental obstacles in extending the work of [Vad04]. First, the construction of [Vad04] made use a computationally unbounded prover in an essential way (as did the previous work on **SZKP**, such as [Oka00]), whereas argument systems are rather unnatural with unbounded provers and hence are typically defined with respect to efficient provers. Second, at the time we did not know of a construction of statistical zero-knowledge arguments for **NP** from any one-way function, which is necessary to make use of the one-way functions constructed from the "J instances" in an SZKP–OWF TRIPLET. (This is clear when trying to characterize **SZKA** but it also turns out to be important for characterizing **CZKA**.) Fortunately, both of these obstacles have been recently overcome in [NV06] and [NOV06], respectively.

2 Preliminaries

If X is a random variable taking values in a finite set $\mathcal{U}$, then we write $x \leftarrow X$ to indicate that x is selected according to X. If S is a subset of $\mathcal{U}$, then $x \leftarrow S$ means that x is selected according to the uniform distribution on S. We adopt the convention that when the same random variable occurs several times in an expression, they refer to a single sample. For example, $\Pr[f(X) = X]$ is defined to be the probability that when $x \leftarrow X$, we have $f(x) = x$. We write U_n to denote the random variable distributed uniformly over $\{0,1\}^n$.

A function $\mu : \mathbb{N} \to [0,1]$ is called *negligible* if $\mu(n) = n^{-\omega(1)}$. We let $\mathrm{neg}(n)$ denote an arbitrary negligible function (i.e., when we say that $f(n) < \mathrm{neg}(n)$ we mean that *there exists* a negligible function $\mu(n)$ such that for every n, $f(n) < \mu(n)$). Likewise, $\mathrm{poly}(n)$ denotes an arbitrary function $f(n) = n^{O(1)}$.

PPT refers to probabilistic algorithms (i.e., Turing machines) that run in *strict* polynomial time. A *nonuniform* PPT algorithm is a pair $(A, \bar{z})$, where $\bar{z} = z_1, z_2, \ldots$ is an infinite sequence of strings where $|z_n| = \mathrm{poly}(n)$, and A is a PPT algorithm that receives pairs of inputs of the form $(x, z_{|x|})$. (The string z_n is the called the *advice string* for A for inputs of length n.) Nonuniform PPT algorithms are equivalent to (nonuniform) families of polynomial-sized Boolean circuits.

Statistical Difference. The *statistical difference* (a.k.a. *variation distance*) between random variables X and Y taking values in $\mathcal{U}$ is defined to be $\Delta(X,Y) \stackrel{\mathrm{def}}{=} \max_{S \subset \mathcal{U}} |\Pr[X \in S] - \Pr[Y \in S]|$. We say that X and Y are *ε-close* if $\Delta(X,Y) \leq \varepsilon$. Conversely, we say that X and Y are *ε-far* if $\Delta(X,Y) > \varepsilon$. For basic facts about this metric, see [SV03, Sec 2.3].

Entropy. The *entropy* of a random variable X is $\mathrm{H}(X) = \mathrm{E}_{x \leftarrow X}[\log(1/\Pr[X = x])]$, where here and throughout the paper all logarithms are to base 2. Intuitively, $\mathrm{H}(X)$ measures the amount of randomness in X *on average* (in bits). For jointly distributed random variables X and Y, we define the *conditional entropy of X given Y* to be

$$\mathrm{H}(X|Y) \stackrel{\mathrm{def}}{=} \mathop{\mathrm{E}}_{y \leftarrow Y}[\mathrm{H}(X|_{Y=y})] = \mathop{\mathrm{E}}_{(x,y) \leftarrow (X,Y)}\left[\log \frac{1}{\Pr[X = x | Y = y]}\right] = \mathrm{H}(X,Y) - \mathrm{H}(Y).$$

2.1 Promise Problems

Roughly speaking, a *promise problem* [ESY84] is a decision problem where some inputs are excluded. Formally, a promise problem is specified by two disjoint sets of strings $\Pi = (\Pi_Y, \Pi_N)$, where we call Π_Y the set of YES *instances* and Π_N the set of NO *instances*. Such a promise problem is associated with the following computational problem: given an input that is "promised" to lie in $\Pi_Y \cup \Pi_N$, decide whether it is in Π_Y or in Π_N. Note that languages are a special case of promise problems (namely, a language L over alphabet Σ corresponds to the promise problem $(L, \Sigma^* \setminus L)$). Thus working with promise problems makes our results more general. Moreover, even to prove our results just for languages, it turns out to be extremely useful to work with promise problems along the way.

The *complement* of a promise problem $\Pi = (\Pi_Y, \Pi_N)$ is the promise problem $\overline{\Pi} = (\Pi_N, \Pi_Y)$. The *union* of two promise problems Π and Γ is the promise problem $\Pi \cup \Gamma = (\Pi_Y \cup \Gamma_Y, \Pi_N \cap \Gamma_N)$. The *intersection* of two promise problems Π and Γ is the promise problem $\Pi \cap \Gamma = (\Pi_Y \cap \Gamma_Y, \Pi_N \cup \Gamma_N)$.

Most complexity classes, typically defined as classes of languages, extend to promise problems in a natural way, by translating conditions on inputs in the language to be conditions on YES instances, and conditions on inputs not in the language to be conditions on NO instances. For example, a promise problem Π is in **BPP** if there is a probabilistic polynomial-time algorithm A such that $x \in \Pi_Y \Rightarrow \Pr[A(x) = 1] \geq 2/3$ and $x \in \Pi_N \Rightarrow \Pr[A(x) = 0] \leq 1/3$. All complexity classes in this paper denote classes of promise problems.

We refer the reader to the recent survey of Goldreich [Gol05] for more on the utility and subtleties of promise problems.

2.2 Auxiliary-Input Cryptographic Primitives

It will be very useful for us to work with cryptographic primitives that are parameterized by an additional "auxiliary" input x, and where the security condition will hold only if x is in some particular set I. Indeed, recall that the SZKP–OWF TRIPLET Characterization refers to such a variant of the notion of one-way functions (as captured by the definitions below). We use the terminology and notation for such primitives from [Vad04].

Definition 2.1. An *auxiliary-input function ensemble* is a collection of functions $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}_{x \in \{0,1\}^*}$, where p and q are polynomials. We call $\mathcal{F}$ *polynomial-time computable* (or just *poly-time*), if there is a (deterministic) polynomial-time algorithm F such that for every $x \in \{0,1\}^*$ and $y \in \{0,1\}^{p(|x|)}$, we have $F(x,y) = f_x(y)$.

Definition 2.2 (one-way function on I)**.** An *auxiliary-input one-way function on* I is a poly-time auxiliary-input function ensemble $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ such that for every nonuniform PPT A, there exists a negligible function μ such that for all $x \in I$,

$$\Pr\left[A(x, f_x(U_{p(|x|)})) \in f_x^{-1}(f_x(U_{p(|x|)}))\right] \leq \mu(|x|).$$

(We note that since A is nonuniform, it is not essential that we give it the input x, because it can be hardwired in as advice, but the definition seems more natural as above.) The standard definition of one-way function is obtained by considering $I = \{1^n : n \geq 0\}$ and $p(n) = n$.

Similarly, the notion of computational indistinguishability has an auxiliary-input analogue (which is widely used in the definition of zero knowledge).

Definition 2.3. An *auxiliary-input probability ensemble* is a collection of random variables $\{X_x\}_{x \in \{0,1\}^*}$, where X_x takes values in $\{0,1\}^{p(|x|)}$ for some polynomial p. We call such an ensemble *samplable*

if there is a probabilistic polynomial-time algorithm M such that for every x, the output $M(x)$ is distributed according to X_x.

Definition 2.4. Two auxiliary-input probability ensembles $\{X_x\}$ and $\{Y_x\}$ are *computationally indistinguishable* on $I \subseteq \{0,1\}^*$ if for every nonuniform PPT D, there exists a negligible function μ such that for all $x \in I$,

$$|\Pr[D(x, X_x) = 1] - \Pr[D(x, Y_x) = 1]| \leq \mu(|x|).$$

Similarly, we say that $\{X_x\}$ and $\{Y_x\}$ are *statistically indistinguishable* on $I \subseteq \{0,1\}^*$ if the above is required for all functions D (instead of only nonuniform PPT). Equivalently, $\{X_x\}$ and $\{Y_x\}$ are statistically indistinguishable on I iff X_x and Y_x are $\mu(|x|)$-close for some negligible function μ and all $x \in I$. We write $\approx_c$ and $\approx_s$ to denote computational and statistical indistinguishability, respectively.

Often, we will informally say "X_x and Y_x are computationally indistinguishable when $x \in I$" to mean the ensembles $\{X_x\}$ and $\{Y_x\}$ are computationally indistinguishable on I.

Now we define auxiliary-input distributionally one-way functions, which are functions that are hard for PPT algorithms to invert in a distributional manner—that is, given y it is hard for PPT algorithms to output a random preimage $f^{-1}(y)$. The standard definition of distributionally one-way function is given by [IL89]; here we give the auxiliary-input analogue.

Definition 2.5 (distributionally one-way function on I)**.** An *auxiliary-input distributionally one-way function on I* is a poly-time auxiliary-input function ensemble $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ such that there exists a polynomial $p(\cdot)$ with the property that for every nonuniform PPT A, the ensembles $\{(U_{p(|x|)}, f_x(U_{p(|x|)}))\}$ and $\{(A(f(U_{p(|x|)})), f_x(U_{p(|x|)}))\}$ are $1/p(|x|)$-far for all sufficient long $x \in I$.

Requiring to invert in a distributional manner is a stronger that just finding a preimage, therefore distributionally one-way functions might seem weaker than one-way functions. However, Impagliazzo–Levin [IL90] proved that they are in fact equivalent. Like almost all reductions between cryptographic primitives, this result immediately extends to the auxiliary-input analogue (using the same proof):

Theorem 2.6 ([IL90])**.** *For every set $I \subseteq \{0,1\}^*$, there exists an auxiliary-input one-way function on I if and only if there exists an auxiliary-input distributionally one-way function on I.*

2.3 Zero-Knowledge Protocols—Brief Introduction

Here we give a brief introduction of the definitions of zero knowledge that we use for the benefit of our more experienced readers. For a more detailed introduction with complete definitions, refer to Section 2.4.

In general, we follow the standard definitions of *interactive protocols*, *interactive proofs* and *arguments*, and *zero-knowledge proofs* and *arguments*, as in [Gol01]. The complexity classes that we use are defined as follows:

- **IP** denotes the class of promise problems possessing interactive proof systems.
- **HV-SZKP** and **HV-CZKP** denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge proofs, respectively. Analogously, **HV-SZKA** and **HV-CZKA** denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge *arguments*, respectively.

- **SZKP** and **CZKP** are the classes of promise problems possessing statistical and computational (auxiliary-input) zero-knowledge proofs, respectively. Analogously, **SZKA** and **CZKA** are the classes of promise problems possessing statistical and computational (auxiliary-input) zero-knowledge *arguments*, respectively.

We highlight the following points:

1. (Proof vs. argument systems) Interactive argument systems refer to protocols whose *soundness* condition is *computational*. That is, only nonuniform PPT cheating provers are guaranteed not to be able to convince the verifier of false statements except with negligible probability; this is a weaker condition that proof systems, where the soundness condition is required of all cheating provers instead of just nonuniform PPT ones. Hence, we say that proof systems have *statistical soundness*.

2. (Prover complexity) Interactive proofs and interactive arguments, and their zero-knowledge analogues, allow the honest prover to be computationally unbounded prover, unless we specify *efficient prover*, which means a polynomial time honest prover strategy given a witness for membership. It was shown in [NV06] than for problems in **NP**, any zero-knowledge *proof* system with an unbounded prover can be transformed into one with an efficient prover; we will show the same for *argument* systems.

2.4 Zero-Knowledge Protocols—Detailed Introduction

An *interactive protocol* (A, B) consists of two algorithms that compute the *next-message function* of the (honest) parties in the protocol. Specifically, $A(x, a, \alpha_1, \ldots, \alpha_k; r)$ denotes the next message α_{k+1} sent by party A when the common input is x, A's auxiliary input is a, A's coin tosses are r, and the messages exchanged so far are $\alpha_1, \ldots, \alpha_k$. There are two special messages, `accept` and `reject`, which immediately halt the interaction. We say that party A (resp. B) is *probabilistic polynomial time (PPT)* if its next-message function can be computed in polynomial time (in $|x| + |a| + |\alpha_1| + \cdots + |\alpha_k|$). Sometimes (though not in this section) we will refer to protocols with a joint output; such an output is specified by a deterministic, polynomial-time computable function of the messages exchanged.

For an interactive protocol (A, B), we write $(A(a), B(b))(x)$ to denote the random process obtained by having A and B interact on common input x, (private) auxiliary inputs a and b to A and B, respectively (if any), and independent random coin tosses for A and B. We call (A, B) *polynomially bounded* if there is a polynomial p such that for all x, a, b, the total length of all messages exchanged in $(A(a), B(b))(x)$ is at most $p(|x|)$ with probability 1. Moreover, if B^* is any interactive algorithm, then A will immediately halt and reject in $(A(a), B^*(b))(x)$ if the total length of the messages ever exceeds $p(|x|)$, and similarly for B interacting with any A^*.

We write $\mathsf{view}_A(A(a), B(b))(x)$ to denote A's view of the interaction, that is a transcript $(x, \gamma_1, \gamma_2, \ldots, \gamma_t; r)$, where the γ_i's are all the messages exchanged and r is A's coin tosses. (Similarly, we define $\mathsf{view}_B(A(a), B(b))(x)$ to denote B's view of the interaction.) When dealing with interactive protocol (P, V), we also write $\langle P, V\rangle(x)$ to denote V's view of the interaction, that is $\langle P, V\rangle(x) = \mathsf{view}_V(P, V)(x)$. Let $\mathsf{transcript}(A(a), B(b))(x)$ denote the messages exchanged in the protocol including the common input x, i.e., $(x, \gamma_1, \gamma_2, \ldots, \gamma_t)$.

The number of *rounds* in an execution of the protocol is the *total* number of messages exchanged between A and B, not including the final `accept`/`reject` message. We call the protocol (A, B) *public coin* if all of the messages sent by B are simply the output of its coin-tosses (independent of the

history), except for the final `accept`/`reject` message which is computed as a deterministic function of the transcript. (Such protocols are also sometimes known as *Arthur-Merlin games* [BM88].)

Definition 2.7 (interactive proofs)**.** An interactive protocol (P, V) is an *interactive proof system* for a promise problem Π if are functions $c, s : \mathbb{N} \to [0, 1]$ such that $1 - c(n) > s(n) + 1/\text{poly}(n)$ and the following holds:

- (Efficiency) (P, V) is polynomially bounded, and V is computable in probabilistic polynomial time.
- (Completeness) If $x \in \Pi_{\text{Y}}$, then V accepts in $(P, V)(x)$ with probability at least $1 - c(|x|)$,
- (Soundness) If $x \in \Pi_{\text{N}}$, then for every P^*, V accepts in $(P^*, V)(x)$ with probability at most $s(|x|)$.

We call $c(\cdot)$ the *completeness error* and $s(\cdot)$ the *soundness error*. We say that (P, V) has *negligible error* if both c and s are negligible. We say that it has *perfect completeness* if $c = 0$. We denote by **IP** the class of promise problems possessing interactive proof systems. We denote **MA** to be the class of promise problems possessing single-round interactive proof systems; that is, the prover P just sends a single message to V, and V uses the prover's message and its own random coins in deciding whether to accept or reject.

We can think of **MA** as a generalization of **NP** where the verification of witnesses is probabilistic. An equivalent definition of **IP** is the class of problems possessing public-coin interactive proof systems with perfect completeness and negligible soundness error [GS89, FGM$^+$89].

Definition 2.8 (interactive arguments)**.** We say that (P, V) is an *interactive argument system* for Π if the soundness condition in Definition 2.7 holds against all nonuniform PPT P^*, instead of every (computationally unbounded) P^*. Specifically, we require both the efficiency and completeness conditions in Definition 2.7 to hold, and the new (weaker) soundness condition is as follows:

- (Soundness) If $x \in \Pi_{\text{N}}$, then for every *nonuniform PPT* P^*, V accepts in $(P^*, V)(x)$ with probability at most $s(|x|)$.

We denote by **IA** the class of promise problems possessing interactive argument systems.

Unlike interactive proofs, the complexity-theoretic characterization of **IA** is not well-studied. In particular, we do not know if general interactive arguments can be made to have public coin or to have perfect completeness. The completeness and soundness error, however, can be made negligibly small by sequential repetition.

There are various notions of zero knowledge, referring to how rich a class of verifier strategies are considered. The weakest is to consider only the "honest verifier," the one that follows the specified protocol.[5]

Definition 2.9 (honest-verifier zero knowledge)**.** An interactive proof system (P, V) for a promise problem Π is *statistical (resp. computational) honest-verifier zero knowledge* if there exists a probabilistic polynomial-time *simulator* S such that the ensembles $\{\langle P, V\rangle(x)\}$ and $\{S(x)\}$ are statistically (resp. computationally) indistinguishable on Π_{Y}.

[5]This is an instantiation of what is called an "honest-but-curious adversary" or "passive adversary" in the literature on cryptographic protocols.

HV-SZKP and **HV-CZKP** denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge proofs, respectively. Analogously, **HV-SZKA** and **HV-CZKA** denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge *arguments*, respectively.

While honest-verifier zero knowledge is already a nontrivial and interesting notion, cryptographic applications usually require that the zero-knowledge condition holds even if the verifier deviates arbitrarily from the specified protocol. This is captured by the following definition.

Definition 2.10 (auxiliary-input zero knowledge)**.** [6] An interactive proof system (P, V) for a promise problem Π is statistical (resp. computational) *(auxiliary-input) zero knowledge* if for every PPT V^* and polynomial p, there exists a PPT S such that the ensembles

$$\{\langle P, V^*(z)\rangle(x)\} \qquad \text{and} \qquad \{S(x,z)\} \tag{1}$$

are statistically (resp. computationally) indistinguishable on the set $\{(x,z) : x \in \Pi_Y, |z| = p(|x|)\}$.

SZKP and **CZKP** are the classes of promise problems possessing statistical and computational (auxiliary-input) zero-knowledge proofs, respectively. Analogously, **SZKA** and **CZKA** are the classes of promise problems possessing statistical and computational (auxiliary-input) zero-knowledge *arguments*, respectively.

The auxiliary input z in the above definition allows one to model a priori information that the verifier may possess before the interaction begins, such as from earlier steps in a larger protocol in which the zero-knowledge proof is being used or from prior executions of the same zero-knowledge proof. As a result, auxiliary-input zero knowledge is closed under sequential composition. That is, if an auxiliary-input zero-knowledge proof is repeated polynomially many times sequentially, then it remains auxiliary-input zero knowledge [GO94]. Plain zero knowledge (i.e., without auxiliary inputs) is not closed under sequential composition [GK96], and thus auxiliary-input zero knowledge is the definition typically used in the literature. In the rest of the paper, we will often drop the word "auxiliary-input" in reference to auxiliary-input zero knowledge.

Typically, a protocol is proven to be zero knowledge by actually exhibiting a single, universal simulator that simulates an arbitrary verifier strategy V^* by using V^* as a subroutine. That is, the simulator does not depend on or use the code of V^* (or its auxiliary input), and instead only requires black-box access to V^*. This type of simulation is formalized as follows.

Definition 2.11 (black-box zero knowledge)**.** An interactive proof system (P, V) for a promise problem Π is statistical (resp. computational) *black-box zero knowledge* if there exists an oracle PPT S such that for every nonuniform PPT V^*, the ensembles

$$\{\langle P, V^*\rangle(x)\}_{x\in\Pi_Y} \qquad \text{and} \qquad \{S^{V^*(x,\cdot;\cdot)}(x)\}_{x\in\Pi_Y}$$

are statistically (resp. computationally) indistinguishable.

[6] Our formulation of auxiliary-input zero knowledge is slightly different than, but equivalent to, the definition in the textbook [Gol01]. We allow V^* to run in polynomial time in the lengths of both its input x and its auxiliary input z, but put a polynomial bound on the length of the auxiliary input. In [Gol01, Sec 4.3.3], V^* is restricted to run in time that is polynomial in just the length of the input x, and no bound is imposed on the length of the auxiliary input z (so V^* may only be able to read a prefix of z). The purpose of allowing the auxiliary input to be longer than the running time of z is to provide additional nonuniformity to the distinguisher (beyond that which the verifier has); we do this directly by allowing the distinguisher to be nonuniform in Definition 2.4.

Even though the above definition does not explicitly refer to an auxiliary input, the definition encompasses auxiliary-input zero knowledge because we allow V^* to be nonuniform (and thus the auxiliary input can be hardwired in V^* as advice). The work of Barak [Bar01] demonstrated that non-black-box zero-knowledge arguments can achieve properties (such as simultaneously being public coin, having a constant number of rounds, and having negligible error) that were known to be impossible for black-box zero knowledge [GK96]. Nevertheless, our results will show that, when ignoring efficiency considerations, black-box zero knowledge is as rich as standard, auxiliary-input zero knowledge; for example, every problem in **CZKA** has a black-box zero-knowledge argument system.

Efficient provers. Although we define interactive arguments without restricting the computational resource the honest prover, it is natural to do since the cheating provers are restricted to be PPT. Hence, interactive arguments are most interesting when considering problems in **NP**, because for these problems, we can restrict the honest prover to be PPT given a witness of membership. To formalize this idea, we define witness relations for problems in **NP**.

Let $W \subseteq \{0,1\}^* \times \{0,1\}^*$ be a relation. Let problem $\Pi^W = (\Pi^W_{\mathrm{Y}}, \Pi^W_{\mathrm{N}})$, where $\Pi^W_{\mathrm{Y}} = \{x \mid \exists w \text{ s.t. } (x,w) \in W\}$ and $\Pi^W_{\mathrm{N}} = \overline{\Pi^W_{\mathrm{Y}}}$. For $(x,w) \in W$, we say that w is an **NP**-*witness* for x. Recall that the class **NP** is the class of problems Π such that $\Pi = \Pi^W$ for a relation W that is decidable in time polynomial in the first input (i.e., x). If $\Pi = \Pi^W$ is an **NP** problem then we say that W is an **NP**-*relation* for Π.

To define witness relations for problems in **MA**, we generalize our relation W as follows: Let $W \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be a relation. Define $W_{\mathrm{Y}} = \{(x,w) : \Pr_r[(x,w,r) \in W] \geq 2/3\}$ and $W_{\mathrm{N}} = \{(x,w) : \Pr_r[(x,w,r) \in W] \leq 1/3\}$. Let problem $\Pi^W = (\Pi^W_{\mathrm{Y}}, \Pi^W_{\mathrm{N}})$, where $\Pi^W_{\mathrm{Y}} = \{x \mid \exists w \text{ s.t. } (x,w) \in W_{\mathrm{Y}}\}$ and $\Pi^W_{\mathrm{N}} = \{x \mid \forall w \text{ s.t. } (x,w) \in W_{\mathrm{N}}\}$. For $(x,w) \in W_{\mathrm{Y}}$, we say that w is an **MA**-*witness* for x. The class **MA** is the class of problems Π such that $\Pi = \Pi^W$ for a relation W that is decidable in time polynomial in the first input (i.e., x). If $\Pi = \Pi^W$ is an **MA** problem then we say that W is an **MA**-*relation* for Π.

In an interactive protocol (P,V) for problem $\Pi \in \mathbf{NP}$ (resp. $\Pi \in \mathbf{MA}$), prover P is an *efficient prover* if $P(w)$ is computable by a probabilistic polynomial-time algorithm when $(x,w) \in W$, where W is an **NP**-relation (resp. **MA**-relation) for Π. We note that **MA** has an equivalent formulation as the class of problems having efficient-prover argument systems (cf., [BLV06]). This means defining efficient provers can be done, without loss of generality, for only problems in **MA**.

Remarks on the definitions. Our definitions mostly follow the now-standard definitions of zero-knowledge proofs as presented in [Gol01], but we highlight the following points:

1. (Prover complexity) Interactive proofs and interactive arguments, and their zero-knowledge analogues, allow the honest prover to be computationally unbounded, unless we specify *efficient prover*. It was shown in [NV06] than for problems in **NP** (actually, also **MA**), any zero-knowledge *proof* system with an unbounded prover can be transformed into one with an efficient prover; we will show the same for *argument* systems.

2. (Promise problems) As has been done numerous times before (e.g. [GK93, SV03]), we extend all of the definitions to promise problems $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ in the natural way, i.e., conditions previously required for inputs in the language (e.g. completeness and zero knowledge) are now required for all YES instances, and conditions previously required for inputs not in the language (e.g., soundness) are now required for all NO instances. Similarly, all of our complexity classes

(e.g., **CZKA**, **SZKP** and **BPP**) are classes of promise problems. These extensions to promise problems are essential for formalizing our arguments, but all the final characterizations and results we derive about **CZKA** automatically hold for the corresponding class of languages, simply because languages are a special case of promise problems.

3. (Nonuniform formulation) As has become standard, we have adopted a nonuniform formulation of zero knowledge, where the computational indistinguishability has to hold even with respect to nonuniform distinguishers and is universally quantified over all YES instances. Uniform treatments of zero knowledge are possible (see [Gol93] and [BLV06, Apdx. A]), but the definitions are much more cumbersome. We do not know whether analogues of our results hold for uniform zero knowledge, and leave that as a problem for future work.

4. (Strict polynomial-time simulators) In this version, we restrict our attention to zero knowledge with respect to simulators that run in *strict* polynomial time. In fact, our techniques actually imply an equivalence between defining the zero-knowledge classes (e.g., **CZKA** and **HV-CZKA**) in terms of expected vs. strict polynomial-time simulators. (This equivalence is achieved following a similar line of reasoning as [Vad04].) Details are deferred to the final version of the paper.

2.5 1-out-of-2-Binding Commitments

We now introduce the notion of $\binom{2}{1}$-binding commitments that will play a central role in establishing our results. These are commitment schemes with two *sequential* and *related* stages such that in each stage, the sender commits to and reveals a value.

Definition 2.12. A *2-phase commitment scheme* (S, R), with security parameter n, consists of four interactive protocols: (S_c^1, R_c^1) the first commitment stage, (S_r^1, R_r^1) the first reveal stage, (S_c^2, R_c^2) the second commitment stage, and (S_r^2, R_r^2) the second reveal stage. For us, both reveal phases will always be noninteractive, consisting of a single message from the sender to the receiver. Throughout, both parties receive the security parameter 1^n as input.

1. In the first commitment stage, S_c^1 receives a private input $\sigma^{(1)} \in \{0,1\}$ and a sequence of coin tosses r_S. At the end, S_c^1 and R_c^1 receive as common output a commitment $z^{(1)}$. (Without loss of generality, we can assume that $z^{(1)}$ is the transcript of the first commitment stage.)

2. In the first reveal stage, S_r^1 and R_r^1 receive as common input the commitment $z^{(1)}$ and a bit $\sigma^{(1)} \in \{0,1\}$ and S_r^1 receives as private input r_S. At the end, S_r^1 and R_r^1 receive a common output τ. (Without loss of generality, we can assume that τ is the concatenation of $z^{(1)}$ and the transcript of first reveal stage, and includes R_r^1's decision to accept or reject.)

3. In the second commitment stage, S_c^2 and R_c^2 both receive the common input $\tau \in \{0,1\}^*$, and S_c^2 receives a private input $\sigma^{(2)} \in \{0,1\}$ and the coin tosses r_S. S_c^2 and R_c^2 receive as common output a commitment $z^{(2)}$. (Without loss of generality, we can assume that $z^{(2)}$ is the concatenation of τ and the transcript of the second commitment stage.)

4. In the second reveal stage, S_r^2 and R_r^2 receive as common input the commitment $z^{(2)}$ and a bit $\sigma^{(2)} \in \{0,1\}$, and S_r^2 receives as private input r_S. At the end, R_r^2 accepts or rejects.

We require that the 2-phase commitment scheme (S, R) satisfies the following two properties.

- *(Perfect Correctness)* For all $\sigma^{(1)}, \sigma^{(2)} \in \{0,1\}$, if both S and R are honest, then R accepts in both phase with probability 1.
- $S = (S^1, S^2) = ((S^1_c, S^1_r), (S^2_c, S^2_r))$ and $R = (R^1, R^2) = ((R^1_c, R^1_r), (R^2_c, R^2_r))$ are computable in probabilistic polynomial time.

We say that (S, R) is *public coin* if it is public coin for R.

Note that instead of providing S with decommitment values as private outputs of the commitment phases, we simply provide it with the same coin tosses throughout (so it can recompute any private state from the transcripts of the previous phases).

As for standard commitment schemes, we define the security of the sender in terms of a hiding property. Loosely speaking, the hiding property for a 2-phase commitment scheme says that *both* commitment phases are hiding. Note that since the phases are run sequentially, the hiding property for the second commitment stage is required to hold even given the receiver's view of the first stage.

Definition 2.13 (hiding)**.** 2-phase commitment scheme (S, R), with security parameter n, is *statistically hiding* if for all adversarial receiver R^*,

1. The views of R^* when interacting with the sender in the first phase on any two messages are statistically indistinguishable. That is, for all $\sigma^{(1)}, \widetilde{\sigma}^{(1)} \in \{0,1\}$,

$$\left\{\mathsf{view}_{R^*}(S^1_c(\sigma^{(1)}), R^*)(1^n)\right\}_{n\in\mathbb{N}} \approx_s \left\{\mathsf{view}_{R^*}(S^1_c(\widetilde{\sigma}^{(1)}), R^*)(1^n)\right\}_{n\in\mathbb{N}}.$$

2. The views of R^* when interacting with the sender in the second phase are statistically indistinguishable no matter what the sender committed to in the first phase. That is, for all $\sigma^{(1)}, \sigma^{(2)}, \widetilde{\sigma}^{(2)} \in \{0,1\}$,

$$\left\{\mathsf{view}_{R^*}(S^2_c(\sigma^{(2)}), R^*)(\Lambda, 1^n)\right\}_{n\in\mathbb{N}} \approx_s \left\{\mathsf{view}_{R^*}(S^2_c(\widetilde{\sigma}^{(2)}), R^*)(\Lambda, 1^n)\right\}_{n\in\mathbb{N}},$$

   where $\Lambda = \mathsf{transcript}(S^1(\sigma^{(1)}), R^*)(1^n)$.

We define *computationally hiding* is an analogous manner, relaxing statistical indistinguishability to require only computational indistinguishability for the receiver's view in the two phases.

We stress that the second condition of the above hiding definition (Definition 2.13) requires that the view of receiver in the second phase be indistinguishable for any two messages even given the transcript of the first phase, $\Lambda = \mathsf{transcript}(S^1(\sigma^{(1)}), R^*)(1^n)$.

Loosely speaking, the binding property says that *at least one* of the two commitment phases is binding. In other words, for every sender S^*, there is at most one "bad" phase $j \in \{1,2\}$ such that given a commitment $z^{(j)}$, S^* can open $z^{(j)}$ successfully both as $\sigma^{(1)}$ and $\widetilde{\sigma}^{(1)} \neq \sigma$ with nonnegligible probability. Actually, we allow this bad phase to be determined dynamically by S^*. Moreover, we require that the second phase be *statistically* binding if the sender breaks the first phase.

Definition 2.14 (1-out-of-2-binding)**.** 2-phase commitment scheme (S, R), with security parameter n, is *computationally* $\binom{2}{1}$*-binding* if there exist a set $\mathcal{B}$ of first-phase transcripts and a negligible function $\varepsilon(n)$ such that:

1. For every (even unbounded) sender S^*, the first-phase transcripts in $\mathcal{B}$ make the second phase statistically binding. That is, for all S^* and every $\tau \in \mathcal{B}$, with probability at least $1 - \varepsilon(n)$ over $z^{(2)} = (S^*, R_c^2)(\tau)$, there is at most one value $\sigma^{(2)} \in \{0,1\}$ such that S^* can convince R_r^2 to accept in the second reveal stage on common input $z^{(2)}$ and $\sigma^{(2)}$.

2. $\forall$ nonuniform PPT S^*,[7] S^* succeeds in the following game with probability at most $\varepsilon(n)$ for all sufficiently large n:

   (a) S^* and R_c^1 interact and output a first-phase commitment $z^{(1)}$.

   (b) S^* outputs two full transcripts τ and $\widetilde{\tau}$ of *both* phases with the following three properties:
   - Transcripts τ and $\widetilde{\tau}$ both start with prefix $z^{(1)}$.
   - The transcript τ contains a successful opening of $z^{(1)}$ to the value $\sigma^{(1)} \in \{0,1\}$ using a first-phase transcript not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in τ.
   - The transcript $\widetilde{\tau}$ contains a successful opening of $z^{(1)}$ to the value $\widetilde{\sigma}^{(1)} \in \{0,1\}$ using a first-phase transcript not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in $\widetilde{\tau}$.

   (c) S^* succeeds if all of the above conditions hold and $\sigma^{(1)} \neq \widetilde{\sigma}^{(1)}$.

We define *statistically* $\binom{2}{1}$*-binding* is an analogous manner, requiring that Condition 2 holds for all (even computationally unbounded) adversarial senders S^* instead of just efficient (nonuniform PPT) ones.

3 Main Results

In this section, we elaborate upon the SZKP–OWF Triplet Characterization Theorem (Thm. 1.4). Specifically, we state four theorems giving a variety of equivalent characterizations of the classes **SZKP**, **CZKP**, **CZKA**, and **SZKA**. The ones for **CZKA** and **SZKA** are new; the others follow from previous work, but are given for comparison. In addition to establishing Theorem 1.4 and hence Theorem 1.2, these theorems show an equivalence between problems having only honest-verifier zero-knowledge protocols, problems having an SZKP–OWF Triplet, and problems with (malicious-verifier) zero-knowledge protocols having desirable properties like an efficient prover, perfect completeness, public coins, and black-box simulation. We note that these characterizations refer only to the classes of problems, but do not necessarily preserve other efficiency measures like round complexity.

For ease of reference, we restate the SZKP–OWF Triplet characterization—Definition 1.3 in the Introduction—below.

Definition 3.1 (SZKP–OWF Triplet Characterization, Restatement of Definition 1.3)**.** Let $\Pi = (\Pi_Y, \Pi_N)$ be a promise problem. We say that (Π, I, J) is a SZKP–OWF Triplet if the following three conditions hold:

1. The promise problem $(\Pi_Y \setminus I, \Pi_N \setminus J)$ is in **SZKP**.

[7] Definitions of cryptographic primitives in the literature often use the reverse order of quantifiers, asking that for every (nonuniform) PPT adversary S^*, there exists a negligible function $\varepsilon(n)$ such that the success probability of S^* is at most $\varepsilon(n)$. However, the two resulting definitions turn out to be equivalent [Bel02].

2. There exists an auxiliary input one-way function $\mathcal{F}$ on I.

3. There exists an auxiliary input one-way function $\mathcal{G}$ on J.

The following two previously known theorems give unconditional characterizations of zero-knowledge *proofs*.

Theorem 3.2 (**SZKP** Characterization Theorem, [Oka00, GSV98, NV06])**.** *For every problem* $\Pi \in \mathbf{IP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{SZKP}$.
2. $\Pi \in \mathbf{HV\text{-}SZKP}$.
3. $(\Pi, \emptyset, \emptyset)$ *is a* SZKP–OWF Triplet.
4. Π *has a statistical zero-knowledge proof system with a black-box simulator, public coins, and perfect completeness. Furthermore, if* $\Pi \in \mathbf{NP}$*, the proof system has an efficient prover.*

Theorem 3.3 (**CZKP** Characterization Theorem, [Vad04, NV06])**.** *For every problem* $\Pi \in \mathbf{IP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{CZKP}$.
2. $\Pi \in \mathbf{HV\text{-}CZKP}$.
3. *There exists a set* $I \subseteq \Pi_{\mathrm{Y}}$ *such that* $(\Pi, I, \emptyset)$ *is a* SZKP–OWF Triplet.
4. Π *has a computational zero-knowledge proof system with a black-box simulator, public coins, and perfect completeness. Furthermore, if* $\Pi \in \mathbf{NP}$*, the proof system has an efficient prover.*

We give analogous characterizations for zero-knowledge *arguments*:

Theorem 3.4 (**CZKA** Characterization Theorem)**.** *For every problem* $\Pi \in \mathbf{NP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{CZKA}$.
2. $\Pi \in \mathbf{HV\text{-}CZKA}$.
3. *There exists sets* $I \subseteq \Pi_{\mathrm{Y}}$ *and* $J \subseteq \Pi_{\mathrm{N}}$ *such that* (Π, I, J) *is a* SZKP–OWF Triplet.
4. Π *has a computational zero-knowledge argument system with a black-box simulator, public coins, perfect completeness, and an efficient prover.*

Theorem 3.5 (**SZKA** Characterization Theorem)**.** *For every problem* $\Pi \in \mathbf{NP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{SZKA}$.
2. $\Pi \in \mathbf{HV\text{-}SZKA}$.
3. *There exists a set* $J \subseteq \Pi_{\mathrm{N}}$ *such that* $(\Pi, \emptyset, J)$ *is a* SZKP–OWF Triplet.
4. Π *has a statistical zero-knowledge argument with a black-box simulator, public coins, perfect completeness, and an efficient prover.*

We prove Theorems 3.4 and 3.5 in Section 4.4. Notice that in these theorems involving zero-knowledge arguments, we have restricted Π to be in **NP** (in contrast to Theorems 3.2 and 3.3, which only restrict to **IP**). The reason for this is that argument systems are mainly interesting when the honest prover runs in polynomial time given a witness for membership (otherwise the protocol would not even be sound against prover strategies with the same resources as the honest prover), and such efficient provers only make sense for problems in **NP** (or actually, **MA**, to which our results generalize easily).[8] In fact our theorems above show that for problems in **NP**, a zero-knowledge protocol without an efficient prover can be converted into one with an efficient prover (by the equivalence of Items 1 and 4 in Theorems 3.2 to 3.5 above).

4 Characterization of Zero-Knowledge Protocols

The steps providing unconditional characterizations of zero-knowledge protocols are as follows.

1. We show that every problem possessing a (honest-verifier) zero-knowledge protocols satisfies the SZKP–OWF Triplet characterization. Depending on the zero knowledge and soundness guarantee, the types of SZKP–OWF Triplet characterization will differ (in whether the sets I and J of one-way function instances are empty or nonempty). This extends the unconditional characterization work of [Vad04] from zero-knowledge proof systems to the more general zero-knowledge argument systems, and is in Section 4.1.

2. Next, we show that every problem satisfying the SZKP–OWF Triplet characterization yields a type of *instance-dependent commitment scheme*. This is based on the techniques of [NOV06, NV06], and is in Section 4.2.

3. Finally, we show that every problem in **NP** having the same type of instance-dependent commitment scheme allow us construct zero-knowledge argument systems with desirable properties like perfect completeness, black-box zero knowledge, public coin, and efficient prover. This is also based on the techniques of [NOV06, NV06], and is in Section 4.3.

4.1 From Zero-Knowledge Protocols to SZKP–OWF Triplet Characterization

In this section, we show that problems possessing (honest verifier) zero-knowledge arguments satisfy the SZKP–OWF Triplet characterization. Specifically, we prove that for every problem Π having a zero-knowledge argument, there exists sets $I \subseteq \Pi_{\mathrm{Y}}$ and $J \subseteq \Pi_{\mathrm{N}}$ such that (Π, I, J) is a SZKP–OWF Triplet. The main difference from [Vad04] is that [Vad04] characterizes only zero-knowledge proofs and has $J = \emptyset$. Intuitively, this set J that we use in our SZKP–OWF Triplet characterization represents the "statistical imperfection" of some NO instances due to the computational soundness property of arguments as compared to statistical soundness of proofs.

Lemma 4.1 (SZKP–OWF Triplet Characterization of **HV-CZKA** and **HV-SZKA**). *If problem Π is in* **HV-CZKA**, *then there exists sets $I \subseteq \Pi_{\mathrm{Y}}$ and $J \subseteq \Pi_{\mathrm{N}}$ such that (Π, I, J) is a* SZKP–OWF Triplet. *Moreover, for $\Pi \in$* **HV-SZKA**, *we can take $J = \emptyset$.*

[8] In fact, we suspect that our results can be generalized to any problem with a public-coin interactive argument system, in particular including all of **IP**, and we will explore this extension for the final version of the paper. A motivation for this is that there are interesting argument systems where the honest prover is not efficient in the usual sense of having a fixed polynomial running time, e.g. the CS proofs and universal arguments of [Mic94, BG02].

Proof Idea. Fix an instance x of the problem $\Pi \in$ **HV-CZKA**. We will now describe on an intuitive level how we determine whether or not to place x in the sets I and J. From the simulator S on input x, we define a simulation-based prover P_S and a simulation-based verifier V_S. On a high level, P_S replies with the same conditional probability as the prover in the output of S, and V_S sends its messages with the same conditional probability as the verifier in the output of S. Now we make the following observations:

1. The interaction between P_S and V_S is identical to the output of the simulator S, on every x.
2. By the zero-knowledge condition, we have that $\langle P_S, V_S \rangle$ is computationally indistinguishable from $\langle P, V \rangle$, when $x \in \Pi_{\mathrm{Y}}$.
3. By assuming, without loss of generality, that the simulator always outputs accepting transcripts, we conclude that P_S makes V_S accepts with probability 1, on every x.

Now, we consider a statistical measure of how "similar" V_S is to V (on instance x, when interacting with simulation-based prover P_S). Using this statistical measure (given in the full proof), we define sets I and J as follows:

- I contains instances $x \in \Pi_{\mathrm{Y}}$ for which V_S is *statistically different* from V.
- J contains instances $x \in \Pi_{\mathrm{N}}$ for which V_S is *statistically similar* to V.

Now the proof that this gives a SZKP–OWF Triplet proceed as follows:

1. On I, we have that V_S is statistically different from V. Nevertheless, by the zero-knowledge condition (as noted above), V_S is computationally similar to V. This enables us to construct one-way functions for instances in I, as shown in [Vad04].
2. On J, we have that V_S is statistically similar to V. Combining this with the fact that P_S will always convince V_S to accept (as noted above), we conclude that P_S convinces V to accept with high probability. By computational soundness of (P, V), it must be the case that P_S is not PPT. Using techniques from Ostrovsky [Ost91], this allows us to convert the simulator S into a distributional one-way function f_S.[9] The result of Impagliazzo–Levin [IL90] then allows us to construct one-way functions from distributional one-way functions (even on an instance-by-instance basis, as we require).
3. To see that $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J) \in$ **SZKP**, observe the following: For those YES instances not in I—that is, instances in $\Pi_{\mathrm{Y}} \setminus I$—the simulated verifier V_S is statistically similar to V. And for those NO instances not in J—that is, instances in $\Pi_{\mathrm{N}} \setminus J$—the simulated verifier V_S is statistically different from V. This gap in the statistical properties allows us to reduce promise problem $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J)$ to one of the complete problems for **SZKP** [SV03, GV99, Vad04].

Proof. Let (P, V) be a zero-knowledge argument system for Π, with simulator S. We now proceed as in the proof of [Vad04] and modify our interactive protocol (P, V) to satisfy following (standard) additional properties.

[9] If f_S is not distributionally one-way, then P_S can be made to be efficient, hence contradicting the computational soundness of (P, V). Interestingly, Ostrovsky [Ost91] uses the assumption that f_S is not distributionally one-way to invert the simulator S on the YES instances, and conclude that Π is not "hard-on-average". Although we use similar techniques as [Ost91], we instead invert S on the NO instances to contradict the computational soundness of (P, V).

- The completeness error $c(|x|)$ and soundness error $s(|x|)$ are both negligible. This can be achieved by standard error reduction via (sequential) repetition.
- On every input x, the two parties exchange $2\ell(|x|)$ messages for some polynomial ℓ, with the verifier sending even-numbered messages and sending all of its $r(|x|) \geq |x|$ random coin tosses in the last message. Having the verifier send its coin tosses at the end does not affect soundness because it is after the prover's last message, and does not affect honest-verifier zero knowledge because the simulator is anyhow required to simulate the verifier's coin tosses.
- On every input x, the simulator always outputs *accepting transcripts*, where we call a sequence γ of 2ℓ messages an accepting transcript on x if all of the verifier's messages are consistent with its coin tosses (as specified in the last message), and the verifier would accept in such an interaction.

For a transcript γ, we denote by γ_i the *prefix* of γ consisting of the first i messages. For readability, we often drop the input x from the notation, e.g. using $\ell = \ell(|x|)$, $\langle P, V\rangle = \langle P, V\rangle(x)$, etc. Thus, in what follows, $\langle P, V\rangle_i$ and S_i are random variables representing prefixes of transcripts generated by the real interaction and simulator, respectively, on a specified input x.

We define the *simulation-based prover*, denoted as P_S, as follows: Given an execution prefix γ_{2i}, for $i = 1, 2, \ldots, \ell - 1$, prover P_S responses as follows.

1. If simulator $S(x)$ outputs a transcript that begins with γ_{2i} with probability 0, then P_S replies with a dummy message.
2. Otherwise, P_S replies according with the same conditional probability as the prover in the output of the simulator. That is, it replies with a string β with probability $p_\beta = \Pr\left[S(x)_{2i+1} = \gamma_{2i} \circ \beta | S(x)_{2i} = \gamma_{2i}\right]$.

Following [AH91, PT96, GV99, Vad04], we consider the following quantity:

$$h(x) = \sum_{i=1}^{\ell} \left[\mathrm{H}(S_{2i}(x)) - \mathrm{H}(S_{2i-1}(x))\right], \tag{2}$$

where $\mathrm{H}(\cdot)$ denotes the *Shannon entropy*. (For a random variable X, its Shannon entropy is $\mathrm{H}(X) = \mathrm{E}_{x \leftarrow X}[\log(1/\Pr[X = x])]$.) How close the value of $h(x)$ gets to r is a measure of how close the "simulated verifier" is from the honest verifier (when interacting with P_S). This is because of the following claim.

Claim 4.2 ([AH91, PT96, GV99])**.** *For every x, and every prover strategy P', we have*

$$\sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P', V\rangle_{2i}(x)) - \mathrm{H}(\langle P', V\rangle_{2i-1}(x))\right] = r.$$

The above sum measures the total entropy contributed by the honest verifier's messages, and it is natural that this should equal the number of coin tosses of the honest verifier. (Recall that the honest verifier reveals its coin tosses at the end.)

Now, we define the sets I and J as follows:

$$I = \{x \in \Pi_{\mathrm{Y}} : h(x) < r - 1\}, \text{ and} \tag{3}$$

$$J = \{x \in \Pi_{\mathrm{N}} : h(x) > r - 2\}. \tag{4}$$

The following three claims are implicit in [Vad04]; for completeness we include their proofs in Section A.1.

Claim 4.3. *Problem* $(\Pi_Y \setminus I, \Pi_N \setminus J) \in \mathbf{SZKP}$.

Claim 4.4. *There exists an auxiliary input one-way function on* I.

Claim 4.5. *For* $\Pi \in \mathbf{HV\text{-}SZKA}$*, we can take* $I = \emptyset$.

The main novelty in our analysis is the following claim:

Claim 4.6. *There exists an auxiliary input one-way function on* J.

Proof of Claim. From the definition of J, we have that $h(x) > r-2$ for every $x \in J$. The following lemma will allow us to show that the probability that the simulation-based prover P_S convinces V on any $x \in J$ is nonnegligible.

Lemma 4.7 ([AH91, PT96, GV99])**.** *For every* x*, if* $\delta = \Pr[(P_S, V)(x) = \texttt{accept}]$*, then*

$$h(x) = \sum_{i=1}^{\ell} [\mathrm{H}(S_{2i}(x)) - \mathrm{H}(S_{2i-1}(x))] \leq r - \log\left(\frac{1}{\delta}\right).$$

Thus on J, we have

$$\Pr[(P_S, V)(x) = \texttt{accept}] \geq 1/4, \tag{5}$$

since $h(x) \geq r-2$ on these instances. This will not be enough to contradict the soundness since (P, V) is only computationally sound and P_S is inefficient. Here we use an idea from Ostrovsky [Ost91]. If we can invert the simulator, then P_S's replies can be approximated efficiently. By the computational soundness of (P, V), this is impossible, so the simulator must be a one-way function.

More precisely, we define g_x as follows:

$$g_x(i, \omega) \stackrel{\text{def}}{=} (x, i, S(x; \omega)_i). \tag{6}$$

Note that g_x is a polynomial-time computable function. For a given x, if g_x is *not* "distributionally" one-way (see Definition 2.5), then we can devise an efficient cheating prover strategy, call it $\widetilde{P}$, that *efficiently* "simulates" our simulation-based prover P_S upto negligible statistical error. The way to do this is to feed a given transcript prefix γ_{2j} (after the verifier has responded), into the inversion algorithm of g_x to obtain the simulation-based prover response. In doing so, we contradict the computational soundness property of (P, V). The above idea is captured by following proposition.

Lemma 4.8 ([Ost91])**.** *Let* $\mathcal{G} \stackrel{\text{def}}{=} \{g_x(i, \omega)\}$*, where* $g_x(i, \omega)$ *is as in (6). For every set* $K \subseteq \{0,1\}^*$*, if* $\mathcal{G}$ *is* not *an auxiliary-input distributionally one-way function on* K*, then for every polynomial* $p(\cdot)$*, there exists a probabilistic polynomial-time prover* $\widetilde{P}$ *such that that it can simulate the replies of the simulation-based prover* P_S *within statistical error* $1/p(|x|)$ *for infinitely many* $x \in K$*. In particular,*

$$\left|\Pr[(\widetilde{P}, V)(x) = \texttt{accept}] - \Pr[(P_S, V)(x) = \texttt{accept}]\right| < 1/p(|x|),$$

for infinitely many $x \in K$.

By Lemma 4.8, if $\mathcal{G} = \{g_x\}$ is not distributionally one-way on J, we can take $J = K$ to get $\left|\Pr[(\widetilde{P}, V)(x) = \texttt{accept}] - \Pr[(P_S, V)(x) = \texttt{accept}]\right| < 1/p(|x|)$, for infinitely many $x \in J$. We also know from (5) that $\Pr[(P_S, V)(x) = \texttt{accept}] \geq 1/4$ for all $x \in J$, combining with the above inequality yields

$$\Pr[(\widetilde{P}, V)(x) = \texttt{accept}] > 1/4 - 1/p(|x|) > 1/8,$$

for infinitely many $x \in J$. This contradicts the computational soundness of (P, V). Therefore, g_x must be distributionally one-way on J.

To get a standard one-way function from g_x, we use the work of Impagliazzo–Levin [IL90] that shows that one-way functions can be constructed out of distributionally one-way function. This is stated in Theorem 2.6, but for ease of reference, we restate it in the following lemma.

Lemma 4.9 (Restatement of Theorem 2.6, [IL90])**.** *For every set $K \subseteq \{0,1\}^*$, there exists an auxiliary-input one-way function on K if and only if there exists an auxiliary-input distributionally one-way function on K.*

This establishes Claim 4.6 above. □

Lemma 4.1 follows from Claims 4.3, 4.4, 4.5 and 4.6 above. □

4.2 From SZKP–OWF Triplet Characterization to Instance-Dependent Commitment Schemes

In this section, we show that the SZKP–OWF Triplet characterization for Π can be used to get *instance-dependent commitments* for Π. These instance-dependent commitments differ from [Vad04] and are more similar in flavor to those in [NV06] in that we have an efficient sender, but we pay the price of having a more complicated $\binom{2}{1}$*-binding* commitment schemes.

Roughly speaking, these $\binom{2}{1}$-binding commitment schemes are commitment schemes that have two phases, each consisting of a commit stage and a reveal stage. (Standard commitment schemes have only one phase.) In the first phase, the sender commits to and reveals one value v_1, and subsequently, in the second phase, the sender commits to and reveals a second value v_2. We require that both phases are hiding, but only that one of them is binding. That is, the binding property only requires that with high probability, the sender will be forced to reveal the correct committed value in at least one of the phases (but which of the two phases can be determined dynamically by the malicious sender). We formally define $\binom{2}{1}$-binding commitment schemes in Section 2.5.

What we are doing in this step of conversion—from SZKP–OWF Triplet characterization to instance-dependent commitment schemes—is analogous to what was done in [NV06] for zero-knowledge *proofs*. In particular, they showed that problems possessing zero-knowledge proofs have a commitment scheme—specifically, an instance-dependent collection of 2-phase commitments—that is *statistically binding*. A natural extension to zero-knowledge *arguments*, which we prove, is to obtain commitments that are *computationally binding*. To do this extension, we use the machinery of [NOV06] that yields a statistically hiding and computationally binding collection of 2-phase commitments from any one-way function.

Before proceeding to the instance-dependent commitment characterizations of zero-knowledge protocols, we define the hiding and binding properties of a collection of 2-phase instance-dependent commitment schemes.

Definition 4.10. A *collection of 2-phase commitment schemes* $\mathcal{C} = \{\mathsf{Com}_1, \cdots, \mathsf{Com}_t\}$ is said to be:

- *Statistically (resp. computationally) hiding* if at least one of the commitments in the collection $\mathcal{C}$ is statistically (resp. computationally) hiding.
- *Statistically (resp. computationally) binding* if all the commitments in the collection $\mathcal{C}$ are statistically (resp. computationally) $\binom{2}{1}$-binding.
- *Public coin* if all all the commitments in the collection $\mathcal{C}$ are public coin.

The asymmetry in the hiding and binding definitions above arises from the need in constructing zero-knowledge protocols from collections of 2-phase commitments schemes (cf., [NV06, NOV06]). Next, we characterize problems in terms of instance-dependent commitments.

Definition 4.11 (Instance-Dependent Commitment Characterization)**.** Problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ has a *instance-dependent collection of 2-phase commitments* if there exists a polynomial-time computable algorithm that maps an instance x to a collection of 2-phase commitment schemes $\mathcal{C}_x = \{\mathsf{Com}_{1,x}, \cdots, \mathsf{Com}_{t,x}\}$ such that:

$$\begin{aligned} x \in \Pi_{\mathrm{Y}} &\Rightarrow \mathcal{C}_x \text{ is hiding;} \\ x \in \Pi_{\mathrm{N}} &\Rightarrow \mathcal{C}_x \text{ is binding.} \end{aligned}$$

We will explicitly specify whether the hiding and binding properties are statistical or computational unless it is clear from context. In addition, the instance-dependent collection of 2-phase commitments is said to be *public coin* if $\mathcal{C}_x$ is public coin for all $x \in \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$.

We now state the characterization of problems possessing zero-knowledge proofs from [NV06], and extend them to zero-knowledge arguments. Keep in mind that problems Π having zero-knowledge proofs imply there exists a set $I \subseteq \Pi_{\mathrm{N}}$ such that $(\Pi, I, \emptyset)$ is a SZKP–OWF Triplet (cf., Theorems 3.2 and 3.3), whereas problems Π having zero-knowledge arguments imply there exists sets $I \subseteq \Pi_{\mathrm{Y}}$ and $J \subseteq \Pi_{\mathrm{N}}$ such that (Π, I, J) is a SZKP–OWF Triplet (cf., Lemma 4.1).

Lemma 4.12 (Commitment Characterization of **CZKP** and **SZKP**, [NV06][10])**.** *Let* $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ *be any problem. If there exists a set* $I \subseteq \Pi_{\mathrm{N}}$ *such that* $(\Pi, I, \emptyset)$ *is a* SZKP–OWF Triplet*, then* Π *has a instance-dependent collection of 2-phase commitments that is computationally hiding,* statistically binding *and public coin. Moreover, if* $I = \emptyset$*, then the collection of commitments is statistically hiding.*

Our extension to argument is the following:

Lemma 4.13 (Commitment Characterization of **CZKA** and **SZKA**)**.** *Let* $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ *be any problem. If there exists sets* $I \subseteq \Pi_{\mathrm{Y}}$ *and* $J \subseteq \Pi_{\mathrm{N}}$ *such that* (Π, I, J) *is a* SZKP–OWF Triplet*, then* Π *has a instance-dependent collection of 2-phase commitments that is computationally hiding,* computationally binding *and public coin. Moreover, if* $I = \emptyset$*, then the collection of commitments is statistically hiding.*

[10] Although [NV06, Theorem 2.4] states the computational analogue by requiring $\Pi \in$ **CZKP**, they actually proved the implication for all problems Π with the property that there exists a set $I \subseteq \Pi_{\mathrm{N}}$ with $(\Pi, I, \emptyset)$ being a SZKP–OWF Triplet, which is in fact equivalent to saying that Π is in **CZKP** for problems $\Pi \in$ **IP** [Vad04].

To prove Lemma 4.13, we note that $\Pi' = (\Pi_Y, \Pi_N \setminus J)$ satisfies the condition of Lemma 4.12, namely there exists a set $I \subseteq \Pi_N$ such that $(\Pi', I, \emptyset)$ is a SZKP–OWF TRIPLET. This gives us a collection of 2-phase commitments that is hiding on Π_Y and statistically binding on $\Pi_N \setminus J$. To extend the binding property of this collection to J, we will need another collection of 2-phase commitments that is binding on J and hiding on Π_Y. We construct this collection from the auxiliary-input one-way function on J using a result from [NOV06] in a way made precise in Lemma 4.14 below. Having these two collections of commitments, we will need to combine these collections in order to obtain a single collection of commitments that is hiding on Π_Y and computationally binding on Π_N; this combination process is made precise in Lemma 4.15.

Lemma 4.14 (implicit in [NOV06])**.** *For every set $J \subseteq \{0,1\}^*$, if there is an auxiliary-input one-way function on J, then problem $(\overline{J}, J)$ has a instance-dependent collection of 2-phase commitments that is statistically hiding, computationally binding and public coin.*

Continuing from our intuition above, we now have two collection of 2-phase commitments; the first collection is hiding on Π_Y, and binding on $\Pi_N \setminus J$, and second collection is hiding on $\overline{J} \supseteq \Pi_Y$ and binding on J. Let problems $\Pi' = (\Pi_Y, \Pi_N \setminus J)$ and $\Gamma' = (\overline{J}, J)$. Observe that what we need is a single collection of 2-phase commitments that is hiding on $\Pi_Y = \Pi'_Y \cap \Gamma'_Y$ and binding on $\Pi_N = \Pi'_N \cup \Gamma'_N$. The next lemma enables us to achieve this.

Lemma 4.15. *If problems $\Pi = (\Pi_Y, \Pi_N)$ and $\Gamma = (\Gamma_Y, \Gamma_N)$ each have instance-dependent collection of 2-phase commitments, then the promise problem $\Pi \cap \Gamma \stackrel{\text{def}}{=} (\Pi_Y \cap \Gamma_Y, \Pi_N \cup \Gamma_N)$ has an instance-dependent collection of 2-phase commitments with the following properties.*

- *If the collections of 2-phase commitments for Π and Γ are both statistically (resp. computationally) hiding, then the collection of 2-phase commitments for $\Pi \cap \Gamma$ is also statistically (resp. computationally) hiding.*
- *If either of the collections of 2-phase commitments for Π and Γ is statistically (resp. computationally) binding, then the collection of 2-phase commitments for $\Pi \cap \Gamma$ is also statistically (resp. computationally) binding.*
- *If the collection of 2-phase commitments for Π and Γ are both public coin, then so is the collection of 2-phase commitments for $\Pi \cap \Gamma$.*

Proof. Let $\mathcal{C}_x = \{\mathsf{Com}_{x,1}, \mathsf{Com}_{x,2}, \cdots, \mathsf{Com}_{x,t}\}$ and $\mathcal{C}'_x = \{\mathsf{Com}'_{x,1}, \mathsf{Com}'_{x,2}, \cdots, \mathsf{Com}'_{x,t'}\}$ be instance-dependent collections of 2-phase commitments for Π and Γ respectively. We will construct a new instance-dependent collection of 2-phase commitments for $\Pi \cap \Gamma$ via the following operation that combines two 2-phase commitments.

> For two 2-phase commitments Com and Com' we can construct a new 2-phase commitment scheme, denoted as $\mathsf{Com} \times \mathsf{Com}'$, that runs Com and Com' in parallel. For each phase, we commit to the same bit for both schemes. That is, to commit to a bit $\sigma \in \{0,1\}$ in the first phase of $\mathsf{Com} \times \mathsf{Com}'$, we commit to σ in the first phases of both Com and Com', running them in parallel. In addition, to commit to some other bit $\lambda \in \{0,1\}$ in the second phase of $\mathsf{Com} \times \mathsf{Com}'$, we do a similar operation—that is, we commit to λ in the second phases of both Com and Com', running them in parallel.

The new instance-dependent collection of 2-phase commitments for $\Pi \cap \Gamma$ is $\mathcal{D}_x = \{\mathsf{Com}_{x,i} \times \mathsf{Com}'_{x,j} : i \in [t], j \in [t']\}$. The following claim regarding the hiding and binding properties of $\mathsf{Com} \times \mathsf{Com}'$ will help us establish our lemma.

Claim 4.16. *For any 2-phase commitments* Com *and* Com', $\mathsf{Com} \times \mathsf{Com}'$ *is:*

1. *Hiding, if both* Com *and* Com' *are hiding.*
2. $\binom{2}{1}$*-binding, if either one of* Com *or* Com' *is* $\binom{2}{1}$*-binding.*

The above claim would imply our lemma because (i) both collections $\mathcal{C}_x$ and $\mathcal{C}'_x$ being hiding, in the sense of Definition 4.10, means that there exists $i \in [t]$ and $j \in [t']$ such that both $\mathsf{Com}_{i,x}$ and $\mathsf{Com}'_{i,j}$ are hiding, and by Claim 4.16, $\mathsf{Com}_{x,i} \times \mathsf{Com}'_{x,j}$ is hiding and hence, $\mathcal{D}_x$ is hiding. And (ii), either of $\mathcal{C}_x$ or $\mathcal{C}'_x$ being binding, in the sense of Definition 4.10, means that either commitments $\mathsf{Com}_{x,i}$'s are $\binom{2}{1}$-binding for all $i \in [t]$, or commitments $\mathsf{Com}'_{x,j}$'s are $\binom{2}{1}$-binding for all $j \in [t']$. In either case, Claim 4.16 gives us that $\mathsf{Com}_{x,i} \times \mathsf{Com}'_{x,j}$ is $\binom{2}{1}$-binding for all $i \in [t]$ and $j \in [t']$, and so $\mathcal{D}_x$ is binding.

> *Proof of Claim.* The hiding property (Item 1), follows from a standard hybrid argument. Intuitively, our new scheme is $\binom{2}{1}$-binding (Item 2) because to break $\mathsf{Com}_i \times \mathsf{Com}'_j$ in both phases, we will have to break both Com_i and Com'_j in both phases (since our new scheme specifies that we have to commit to the same bit for both Com and Com'). Since one of Com or Com' is $\binom{2}{1}$-binding, we have a contradiction.
>
> To formalize this intuition, let the binding sets for Com and Com' be $\mathcal{B}$ and $\mathcal{B}'$, respectively. Assume, without loss of generality, that Com is $\binom{2}{1}$-binding. Our new binding set will be all first-phase transcripts (τ, τ') (recall that we are running Com and Com' in parallel) with $\tau \in \mathcal{B}$. Note that we do not care about if τ' is in $\mathcal{B}'$ or not since Com' has no binding guarantee. It is now clear that any adversary that breaks the $\binom{2}{1}$-binding property of $\mathsf{Com} \times \mathsf{Com}'$ can be easily transformed into an adversary that breaks the $\binom{2}{1}$-binding property of Com by simulating Com' on its own. □

This completes our proof of Lemma 4.15. □

Proof of Lemma 4.13. First, note that $\Pi' = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus J)$ satisfies the condition of Lemma 4.12, namely there exists a set $I \subseteq \Pi_{\mathrm{N}}$ such that $(\Pi', I, \emptyset)$ is a SZKP–OWF Triplet. By Lemma 4.12, Π' has an instance-dependent collection of 2-phase commitment that is computationally hiding and statistically binding. (If $I = \emptyset$, this collection is statistically hiding.)

By Lemma 4.14, we know that $(\overline{J}, J)$ has an instance-dependent collection of 2-phase commitment that is statistically hiding and computationally binding. Applying Lemma 4.15 yields our result as $\Pi_{\mathrm{Y}} = \Pi_{\mathrm{Y}} \cap \overline{J}$, and $\Pi_{\mathrm{N}} = (\Pi_{\mathrm{N}} \setminus J) \cup J$. □

4.3 From Instance-Dependent Commitment Schemes to Zero-Knowledge Protocols

Having obtained collections of 2-phase commitments from the previous section, we now use these commitments to construct zero-knowledge protocols with desirable properties like perfect completeness, black-box zero knowledge, public coin, and efficient prover. The zero-knowledge protocol that we use is similar to [NV06], where they use it to obtain zero-knowledge proof systems. Because we are dealing with argument systems, the analysis that we need follows from [NOV06], which uses the same protocol of [NV06] to obtain zero-knowledge argument systems.

Lemma 4.17 ([NV06, NOV06]). *If problem* $\Pi = (\Pi_Y, \Pi_N) \in \mathbf{NP}$ *has an instance-dependent collection of 2-phase commitments that is statistically (resp. computationally) hiding and computationally binding, then* Π *has a statistical (resp. computational) zero-knowledge argument system* (P, V) *with the following properties.*

1. *Protocol* (P, V) *has an efficient prover.*
2. *Protocol* (P, V) *is black-box zero knowledge.*
3. *Protocol* (P, V) *has perfect completeness.*
4. *Protocol* (P, V) *is public coin if the collection of 2-phase commitments is public coin.*

The above lemma constraints Π to be in **NP**, since it deals with zero knowledge protocols that has an efficient prover. It can be extended to **MA**, which has an equivalent formulation as the class of problems having efficient-prover argument systems (cf., [BLV06]).

We suspect that our results can be extended to all problems having *public coin* interactive argument systems (in particular this includes all of **IP**), and in particular show that the zero-knowledge property never requires a prover of higher complexity than the original, non-zero-knowledge argument systems. We note that there are interesting interactive arguments with provers that are not efficient in the standard sense of running in a fixed polynomial-time—these include CS proofs [Mic94] and universal arguments [BG02]. We will explore such extensions for the final version of the paper.

4.4 Putting It All Together

We now show how our lemmas imply our main theorems from Section 3. We restate Theorems 3.4 and 3.5, adding the Commitment Characterization (cf., Lemmas 4.12 and 4.13) as an equivalent characterization.

Theorem 4.18 (**CZKA** Characterization Theorem, Restatement of Theorem 3.4). *For every problem* $\Pi \in \mathbf{NP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{CZKA}$.
2. $\Pi \in \mathbf{HV\text{-}CZKA}$.
3. *There exists sets* $I \subseteq \Pi_Y$ *and* $J \subseteq \Pi_N$ *such that* (Π, I, J) *is a* SZKP–OWF Triplet.
4. Π *has an instance-dependent collection of 2-phase commitments that is computationally hiding, computationally binding and public coin.*
5. Π *has a public-coin computational zero-knowledge argument with a black-box simulator, perfect completeness, and an efficient prover.*

Theorem 4.19 (**SZKA** Characterization Theorem, Restatement of Theorem 3.5). *For every problem* $\Pi \in \mathbf{NP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{SZKA}$.
2. $\Pi \in \mathbf{HV\text{-}SZKA}$.
3. *There exists a set* $J \subseteq \Pi_N$ *such that* $(\Pi, \emptyset, J)$ *is a* SZKP–OWF Triplet.

4. Π *has an instance-dependent collection of 2-phase commitments that is statistically hiding, computationally binding and public coin.*

5. Π *has a public-coin statistical zero-knowledge argument with a black-box simulator, perfect completeness, and an efficient prover.*

Proof of Theorems 4.18 and 4.19. The implications for both theorems are captured by the same lemmas, so we can conveniently state them together.

(1) $\Rightarrow$ **(2)** and **(5)** $\Rightarrow$ **(1)** is trivial from definition.

(2) $\Rightarrow$ **(3)** is Lemma 4.1.

(3) $\Rightarrow$ **(4)** is Lemma 4.13.

(4) $\Rightarrow$ **(5)** is Lemma 4.17. This is the only step that needs $\Pi \in \mathbf{NP}$.

□

References

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, 2001.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[Bel02] Mihir Bellare. A note on negligible functions. *Journal of Cryptology*, 15(4):271–284, 2002.

[BG02] Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity*, pages 194–203, 2002.

[BLV06] Boaz Barak, Yehuda Lindell, and Salil Vadhan. Lower bounds for non-black-box zero knowledge. *Journal of Computer and System Sciences*, 72(2):321–391, 2006. Extended abstract in *FOCS '04*.

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[ESY84] Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, May 1984.

[FGM+89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 429–442. JAC Press, Inc., 1989.

[GK93] 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.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW91] 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 ACM*, 38(1):691–729, 1991.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[Gol93] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[Gol05] Oded Goldreich. On promise problems (a survey in memory of Shimon Even [1935-2004]). Technical Report TR05–018, Electronic Colloquium on Computational Complexity, February 2005.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989.

[GSV98] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, 1998.

[GV99] Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 54–73, 1999.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

[IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 812–821, 1990.

[Mic94] Silvio Micali. CS proofs. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 436–453, 1994.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[NOV06] Minh Nguyen, Shien Jin Ong, and Salil Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science*, 2006.

[NV06] Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000.

[Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 133–138, 1991.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems*, pages 3–17, 1993.

[PT96] Erez Petrank and Gábor Tardos. On the knowledge complexity of NP. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 494–503, 1996.

[SV03] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003.

[Vad04] Salil Vadhan. An unconditional study of computational zero knowledge. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04)*, pages 176–185, 2004. Full version to appear in *SIAM J. Computing* Special Issue on Randomness & Complexity.

A Characterization of Zero-Knowledge Protocols

A.1 From Zero-Knowledge Protocols to SZKP–OWF Triplet Characterization

We restate and prove Claims 4.3, 4.4 and 4.5 in Section 4.1.

Claim A.1 (Restatement of Claim 4.3)**.** *Problem* $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J) \in \mathbf{SZKP}$.

Before proving the above claim, we first define the *conditional entropy* of two jointly distributed random variables as follows: For jointly distributed random variables X and Y, we define the *conditional entropy of X given Y* to be

$$\mathrm{H}(X|Y) \stackrel{\text{def}}{=} \mathop{\mathrm{E}}_{y \leftarrow Y}[\mathrm{H}(X|_{Y=y})] = \mathop{\mathrm{E}}_{(x,y) \leftarrow (X,Y)}\left[\log \frac{1}{\Pr[X = x | Y = y]}\right] = \mathrm{H}(X,Y) - \mathrm{H}(Y).$$

Proof of Claim. We note the following lemma.

Lemma A.2 ([Vad04])**.** *Consider the problem* Conditional Entropy Approximation $= (\mathrm{CEA}_{\mathrm{Y}}, \mathrm{CEA}_{\mathrm{N}})$*, where* $\mathrm{CEA}_{\mathrm{Y}} = \{((X,Y),r) : \mathrm{H}(X|Y) \geq r\}$ *and* $\mathrm{CEA}_{\mathrm{N}} = \{((X,Y),r) : \mathrm{H}(X|Y) \leq r-1\}$*. Here (X,Y) is a* samplable joint distribution *specified by two circuits that use the same coin tosses.* Conditional Entropy Approximation *is complete for* **SZKP**.

We prove our claim by reducing $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J)$ to Conditional Entropy Approximation. Given input x, we construct circuits that sample from the following (joint) random variables.

(X,Y): Select $i \leftarrow \{1, \ldots, \ell(|x|)\}$, choose random coin tosses $\omega \leftarrow \{0,1\}^{r(|x|)}$ for the simulator, and output $(S_{2i}(x;\omega), S_{2i-1}(x;\omega))$.

When $x \in \Pi_{\mathrm{Y}} \setminus I$, we have $h(x) \geq r-1$ and hence

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) \geq \frac{r-1}{\ell}.$$

And when $x \in \Pi_{\mathrm{N}} \setminus J$, we have have $h(x) \leq r-2$ and we get

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) \leq \frac{r-2}{\ell}.$$

This is what we need to prove, except the entropy gap is only $1/\ell$. This can be increased to 1 by taking ℓ independent samples from the joint distribution. That is, we define $(\overline{X}, \overline{Y}) = ((X_1, \ldots, X_\ell), (Y_1, \ldots, Y_\ell))$, where the (X_i, Y_i)'s are independent copies of (X,Y). Since $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J)$ reduces to Conditional Entropy Approximation, Lemma A.2 gives us that $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus J) \in \mathbf{SZKP}$. □

Claim A.3 (Restatement of Claim 4.4)**.** *There exists an auxiliary input one-way function on I.*

Proof of Claim. We note the following lemma.

Lemma A.4 ([Vad04])**.** *Let $I \subseteq \{0,1\}^*$ be any set. Assume that there exists a polynomial-time computable mapping that maps x to samplable joint distributions (X,Y) and a parameter r such that $\mathrm{H}(X|Y) \leq r-1$, but $\mathrm{H}(X'|Y') \geq r$ for some (X',Y') indistinguishable from (X,Y). Then there exists an auxiliary-input one-way function on I.*

When $x \in \Pi_{\mathrm{Y}}$, then S is computationally indistinguishable from $\langle P,V\rangle$. So (X,Y), as defined in the proof of Claim 4.3 above, is indistinguishable from $(X',Y') = (\langle P,V\rangle_{2L}, \langle P,V\rangle_{2L-1})$, where L denotes a uniform random element of $\{1,\ldots,\ell\}$.

By Claim 4.2, we have:

$$\mathrm{H}(X'|Y') = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(\langle P,V\rangle_{2i}|\langle P,V\rangle_{2i-1}) = \frac{r}{\ell},$$

for all $x \in \Pi_{\mathrm{Y}}$. And when $x \in I \subseteq \Pi_{\mathrm{Y}}$, we have have $h(x) < r-1$ and hence:

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) < \frac{r-1}{\ell}.$$

Again, like in the proof of Claim 4.3, we can increase the entropy gap between $\mathrm{H}(X'|Y')$ and $\mathrm{H}(X|Y)$ to 1. Finally, we apply Lemma A.4 to establish our claim. □

Claim A.5 (Restatement of Claim 4.5)**.** *For $\Pi \in$ **HV-SZKA**, we can take $I = \emptyset$.*

Proof of Claim. For $\Pi \in$ **HV-SZKA**, the output of the simulator $S(x)$ is statistically close to $\langle P,V\rangle(x)$ for every $x \in \Pi_{\mathrm{Y}}$. This implies that $I = \emptyset$, since for every $x \in \Pi_{\mathrm{Y}}$, we have

$$h(x) > \sum_{i=1}^{\ell} [\mathrm{H}(\langle P,V\rangle_{2i}(x)) - \mathrm{H}(\langle P,V\rangle_{2i-1}(x))] - \mathrm{neg}(|x|) = r - \mathrm{neg}(|x|),$$

the last equality following Claim 4.2. □

ECCC ISSN 1433-8092
http://eccc.hpi-web.de/