Zero Knowledge and Soundness are Symmetric*

Shien Jin Ong† Salil Vadhan†

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

March 23, 2007

Abstract

We give a complexity-theoretic characterization of the class of problems in **NP** having zero-knowledge argument systems. This characterization is symmetric in its treatment of the zero knowledge and the soundness conditions, and thus 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.

*An extended abstract of this paper will appear in the *26th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2007)*.

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

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 [GMR], 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 learns nothing from the interaction.[1] Similarly, for soundness, we have *statistical soundness*, giving rise to *proof systems*, where even a computationally unbounded prover cannot convince the verifier of a false statement (except with negligible probability), and *computational soundness*, giving rise to *argument systems* [BCC], 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 like an 'unforgeability' condition. However, in a remarkable paper, Okamoto [Oka] showed that they are actually symmetric in the case of statistical security.

Theorem 1.1 ([Oka, GSV][2])**.** *The class* **SZKP** *of problems having statistical zero-knowledge proofs is closed under complement. That is,* $\Pi \in$ **SZKP** *if and only if* $\overline{\Pi} \in$ **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 a YES instance (that is, when the statement being proven is true) and soundness is required only when x is a NO instance (that is, when 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 when both security conditions are statistical.

We ask whether an analogous theorem holds when the security conditions are *computational*, namely when considering computational zero-knowledge arguments. If we make complexity assumptions, then the answer is yes. Indeed, the classical results of Goldreich, Micali, and Wigderson [GMW], and Brassard, Chaum, and Crépeau [BCC] show that every problem in **NP** has computational zero-knowledge argument systems under widely believed complexity assumptions, and in fact either one of the security conditions can be made statistical. Moreover, it is known

[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 [GSV] it was shown this is the same as the class of languages having general statistical zero-knowledge proofs.

that the existence of one-way functions (OWF) suffices for the construction of computational zero-knowledge proof systems and statistical zero-knowledge argument systems for every problem in **NP** [Nao, HILL, NOV]. Thus, the existence of one-way functions implies that computational zero knowledge and computational soundness are symmetric for problems in $\mathbf{NP} \cap \mathbf{coNP}$, by implying that all problems in $\mathbf{NP} \cap \mathbf{coNP}$ and their complements have computational zero-knowledge arguments. We note that here, and throughout the paper, we usually restrict 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]

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

Theorem 1.2 (Symmetry Theorem)**.**

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

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

Observe 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 Characterization

The Symmetry Theorem is obtained by new characterizations of the classes of problems having zero-knowledge protocols, and moreover these characterizations treat zero knowledge and soundness symmetrically. These characterizations are a generalization of the "SZK/OWF Characterization Theorem" of [Vad], which says that any problem having a computational zero-knowledge *proof* system 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 zero-knowledge *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 a 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 that we consider—for instance, **SZKP**, **CZKP**, **SZKA**, and **CZKA**—generalize to promise problems in a natural way; completeness and zero knowledge are required for YES instances, and soundness is required for NO instances.

Definition 1.3 (SZKP–OWF Condition)**.** We say that promise problem $\Pi = (\Pi_Y, \Pi_N)$ satisfies the SZKP–OWF Condition if there exists a set of instances $I \subseteq \Pi_Y \cup \Pi_N$ such that the following two conditions hold.

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

[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.

- There exists a polynomial-time computable function $f_x\colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$, with $n(\cdot)$ and $m(\cdot)$ being polynomials and instance x given as an auxiliary input, such that for every nonuniform probabilistic polynomial-time adversary A, and for every constant $c > 0$, we have

$$\Pr_{y \leftarrow \{0,1\}^{n(|x|)}} \left[A(f_x(y)) \in f_x^{-1}(f_x(y)) \right] \leq |x|^{-c} \ ,$$

  for every sufficiently long $x \in I$.

We call I the set of OWF *instances*, $I \cap \Pi_{\mathrm{Y}}$ the set of OWF YES *instances*, and $I \cap \Pi_{\mathrm{N}}$ the set of OWF NO *instances*.

We use the SZKP–OWF CONDITION to characterize the classes of problems having zero-knowledge protocols.

Theorem 1.4 (SZKP–OWF Characterization of Zero Knowledge)**.**

1. (**SZKP** [trivial]) *Problem* $\Pi \in$ **IP** *has a statistical zero-knowledge proof system if and only if* Π *satisfies the* SZKP–OWF CONDITION *without* OWF *instances, namely* $I = \emptyset$.

2. (**CZKP** [Vad]) *Problem* $\Pi \in$ **IP** *has a computational zero-knowledge proof system if and only if* Π *satisfies the* SZKP–OWF CONDITION *without* OWF NO *instances, namely* $I \cap \Pi_{\mathrm{N}} = \emptyset$.

3. (**SZKA** [new]) *Problem* $\Pi \in$ **NP** *has a statistical zero-knowledge argument system if and only if* Π *satisfies the* SZKP–OWF CONDITION *without* OWF YES *instances, namely* $I \cap \Pi_{\mathrm{Y}} = \emptyset$.

4. (**CZKA** [new]) *Problem* $\Pi \in$ **NP** *has a computational zero-knowledge argument system if and only if* Π *satisfies the* SZKP–OWF CONDITION.

Theorem 1.2, our Symmetry Theorem between computational zero knowledge and computational soundness, follows directly from: (i) Theorem 1.4 above, (ii) Okamoto's Theorem that **SZKP** is closed under complement (Theorem 1.1), and (iii) the symmetric role played by the set of OWF instances I in the SZKP–OWF CONDITION.

The advantage of the SZKP–OWF Characterization Theorem is that it reduces 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 the case of zero-knowledge *proof* systems [Oka, GSV, Vad, NV], or were known under the complexity assumptions like the existence of one-way functions for the case of zero-knowledge argument systems [GMW, Nao, HILL, NOV].

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

In the forward direction, going from **CZKA** or **SZKA** to an SZKP–OWF CONDITION, we combine the work of [Vad] with an idea of Ostrovsky [Ost] to construct a one-way function on NO

instances in $I \cap \Pi_{\mathrm{N}}$. 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 [OW].) 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 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 CONDITION to **CZKA** or **SZKA**, there were more fundamental obstacles in extending the work of [Vad]. First, the construction of [Vad] made use of a computationally unbounded prover in an essential way (as did the previous work on **SZKP**, such as [Oka]), 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 instances in $I \cap \Pi_{\mathrm{N}}$—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 [NV] and [NOV], respectively.

In more detail, we prove the reverse direction by showing that every problem satisfying the SZKP–OWF CONDITION has an *instance-dependent* commitment scheme[5] [BMO, IOS, MV], and then using techniques from [GMW, IOS], we show that every problem in **NP** with such a commitment scheme has a zero-knowledge argument system. In the original version of this paper [OV], our instance-dependent commitment scheme inherited a certain "1-out-of-2" binding property from [NV] and [NOV]. This property is weaker and more complicated than the standard binding property of commitments, but sufficed for establishing our main theorems (Theorems 1.2 and 1.4). Subsequently, the results of [NV] and [NOV] have been improved to yield standard-binding commitments, the latter by Haitner and Reingold [HR] and the former by [HORV]. Thus in this version, we use standard-binding instance-dependent commitments, as it simplifies our presentation.

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 $\varepsilon : \mathbb{N} \to [0,1]$ is called *negligible* if $\varepsilon(n) = n^{-\omega(1)}$. 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

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

[5] Informally, instance-dependent commitment schemes for a problem Π are commitment schemes where the hiding and binding properties are required to hold only on the YES and NO instances of Π, respectively. A formal definition was first given by Itoh, Ohta, and Shizuya [IOS], and we provide it in Section 2.2.

function $\varepsilon(n)$ such that for every n, $f(n) < \varepsilon(n)$). Likewise, 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 $|z_n| = \text{poly}(n)$, 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 $\Delta(X,Y) = \max_{S \subset \mathcal{U}} |\Pr[X \in S] - \Pr[Y \in S]|$. 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 [SV, Sec 2.3].

2.1 Promise Problems

A *promise problem* [ESY], stated informally, 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 [Gol3] for more on the utility and subtleties of promise problems.

2.2 Instance-Dependent Cryptographic Primitives

It will be very useful for us to work with cryptographic primitives that may depend on an instance x of a problem $\Pi = (\Pi_Y, \Pi_N)$, and where the security condition will hold only if x is in some particular set $I \subseteq \{0,1\}^*$. Indeed, recall that the SZKP–OWF CONDITION (Definition 1.3) refers to such a variant of of one-way functions, as captured by Definition 2.2 below.

Instance-Dependent One-Way Functions. To define instance-dependent one-way functions, we will need to define what it means for a function to be *instance dependent*.

Definition 2.1. An *instance-dependent function* is a family $\mathcal{F} = \{f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}_{x \in \{0,1\}^*}$, where $n(\cdot)$ and $m(\cdot)$ are polynomials. We call $\mathcal{F}$ *polynomial-time computable* if there is a deterministic polynomial-time algorithm F such that for every $x \in \{0,1\}^*$ and $y \in \{0,1\}^{n(|x|)}$, we have $F(x,y) = f_x(y)$.

To simplify notation, we often write $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ to mean the instance-dependent function $\{f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}_{x \in \{0,1\}^*}$.

Definition 2.2 (Instance-Dependent One-Way Function)**.** For any set $I \subseteq \{0,1\}^*$, a polynomial-time computable instance-dependent function $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ is an *instance-dependent one-way function on I* if for every nonuniform PPT adversary A, there exists a negligible function ε such that for every $x \in I$,

$$\Pr_{y \leftarrow \{0,1\}^{n(|x|)}} \left[A(x, f_x(y)) \in f_x^{-1}(f_x(y)) \right] \leq \varepsilon(|x|) \ .$$

Next we consider an instance-dependent variant of *distributionally one-way functions*, which are functions that are hard for PPT adversaries to invert in a distributional manner—that is, given y it is hard for PPT adversaries to output a random preimage $f^{-1}(y)$. The standard definition of distributionally one-way function is given by Impagliazzo and Luby [IL]; here we give the instance-dependent analogue.

Definition 2.3 (Instance-Dependent Distributionally One-Way Function)**.** For any set $I \subseteq \{0,1\}^*$, a polynomial-time computable instance-dependent function $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ is an *instance-dependent distributionally one-way function on I* if there exists a polynomial $p(\cdot)$ such that for every nonuniform PPT adversary A, the random variables $(U_{n(|x|)}, f_x(U_{n(|x|)}))$ and $(A(f_x(U_{n(|x|)})), f_x(U_{n(|x|)}))$ are $1/p(|x|)$-far for all sufficiently long $x \in I$.

Asking to invert in a distributional manner is a stronger requirement that just finding a preimage, therefore distributionally one-way functions might seem weaker than one-way functions. However, Impagliazzo and Luby [IL] proved that they are in fact equivalent. Like almost all reductions between cryptographic primitives, this result immediately extends to the instance-dependent analogue (using the same proof).

Proposition 2.4 (based on [IL, Lemma 1])**.** *For every set $I \subseteq \{0,1\}^*$, there exists an instance-dependent one-way function on I if and only if there exists an instance-dependent distributionally one-way function on I.*

Indistinguishability of Instance-Dependent Ensembles. The notions of statistical and computational indistinguishability have instance-dependent analogues. But first, we define an instance-dependent analogue of probability ensembles.

Definition 2.5. An *instance-dependent probability ensemble* is a collection of random variables $\{A_x\}_{x \in \{0,1\}^*}$, where A_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 A_x.

Definition 2.6. Two instance-dependent probability ensembles $\{A_x\}_{x\in\{0,1\}^*}$ and $\{B_x\}_{x\in\{0,1\}^*}$ are *computationally indistinguishable* on $I \subseteq \{0,1\}^*$ if for every nonuniform PPT D, there exists a negligible function ε such that for all $x \in I$,

$$|\Pr[D(x, A_x) = 1] - \Pr[D(x, B_x) = 1]| \leq \varepsilon(|x|) \ .$$

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

Instance-Dependent Commitment Schemes. Recall that a (standard) commitment scheme is a two-stage protocol between a sender and a receiver. In the first stage, called the *commit stage*, the sender "commits" to a private message m. In the second stage, called the *reveal stage*, the sender reveals m and "proves" that it was the message to which she committed in the first stage. We require two properties of commitment schemes. The *hiding* property says that the receiver learns nothing about m in the commit stage. The *binding* property says that after the commit stage, the sender is bound to a particular value of m; that is, she cannot successfully open the commitment to two different bits in the reveal stage.

Instance dependent analogues of commitments schemes are commitments schemes that are tailored specifically to a specific problem Π. More precisely, *instance-dependent commitment schemes* [BMO, IOS, MV] receive an instance x of the problem Π as auxiliary input, and are required to be hiding when $x \in \Pi_{\mathrm{Y}}$ and be binding when $x \in \Pi_{\mathrm{N}}$. Thus, they are a relaxation of standard commitment schemes, since we do not require that the hiding and binding properties hold at the same time. Nevertheless, as observed in [IOS], this relaxation is still useful in constructing zero-knowledge protocols. The reason is that zero-knowledge protocols based on commitments (for example, the protocol of [GMW]) typically use only the hiding property in proving zero knowledge (which is required only when x is a YES instance) and use only the binding property in proving soundness (which is required only when x is a NO instance).

We give a definition of instance-dependent commitment schemes that extends the standard (that is, non-instance dependent) definition of commitment schemes in a natural way. Note that in our definition below, the reveal stage is *noninteractive* (that is, consisting of a single message from the sender to the receiver). This because in the reveal stage, without loss of generality, we can have the sender provide the receiver the random coin tosses it used in the commit stage, and the receiver verifies consistency.

Definition 2.7 (instance-dependent commitment schemes)**.** An *instance-dependent commitment scheme* is a family $\{\mathsf{Com}_x\}_{x\in\{0,1\}^*}$ with the following properties:

1. Scheme Com_x consists of a commit and a reveal stage. In both stages, the sender and the receiver receive instance x as common input, and hence we denote them as S_x and R_x, respectively, and write $\mathsf{Com}_x = (S_x, R_x)$.

2. At the beginning of the commit stage, sender S_x receives a private input $b \in \{0,1\}$. At the end of the commit stage, both sender S_x and receiver R_x output a commitment c.

3. In the reveal stage, sender S_x sends a pair (b, d), where d is interpreted as the decommitment string for bit b. Receiver R_x accepts or rejects based on x, b, d, and c.

4. The sender S_x and receiver R_x algorithms are computable in polynomial time (in $|x|$), given x as auxiliary input.

5. For every $x \in \{0,1\}^*$, R_x will always accept (with probability 1) if both sender S_x and receiver R_x follow their prescribed strategy.

Instance-dependent commitment scheme $\{\mathsf{Com}_x = (S_x, R_x)\}_{x\in\{0,1\}^*}$ is *public coin* if for every $x \in \{0,1\}^*$, all messages sent by R_x in the commit phase are independent random coins.

To simplify notation, we write Com_x or (S_x, R_x) to denote instance-dependent commitment scheme $\{\mathsf{Com}_x = (S_x, R_x)\}_{x\in\{0,1\}^*}$. Next, we define the hiding and binding properties of instance-dependent commitments.

Definition 2.8 (hiding)**.** Instance-dependent commitment scheme $\mathsf{Com}_x = (S_x, R_x)$ is *statistically [resp., computationally] hiding* on $I \subseteq \{0,1\}^*$ if for every [resp., nonuniform PPT] R^*, the ensembles $\{\text{view}_{R^*}(S_x(0), R^*)\}_{x\in I}$ and $\{\text{view}_{R^*}(S_x(1), R^*)\}_{x\in I}$ are statistically [resp., computationally] indistinguishable, where random variable $\text{view}_{R^*}(S_x(b), R^*)$ denotes the view of R^* in the commit stage interacting with $S_x(b)$.

Definition 2.9 (binding)**.** Instance-dependent commitment scheme $\mathsf{Com}_x = (S_x, R_x)$ is *statistically [resp., computationally] binding* on $I \subseteq \{0,1\}^*$ if for every [resp., nonuniform PPT] S^*, there exists a negligible function ε such that for all $x \in I$, the adversarial sender S^* succeeds in the following game with probability at most $\varepsilon(|x|)$.

> S^* interacts with R_x in the commit stage obtaining commitment c. Then S^* outputs pairs $(0, d_0)$ and $(1, d_1)$, and succeeds if in the reveal stage, $R_x(0, d_0, c) = R_x(1, d_1, c) =$ `accept`.

For a problem $\Pi = (\Pi_Y, \Pi_N)$, we say that instance-dependent commitment scheme Com_x for Π is statistically [resp., computationally] binding on the NO instances if Com_x is statistically [resp., computationally] binding on Π_N.

2.3 Zero-Knowledge Protocols—Brief Introduction

For the benefit of more experienced readers, we briefly recall the variants of zero knowledge that we use. Section 2.4 contains a more detailed introduction with complete definitions. Informal descriptions of the complexity classes used are listed below.

- **IP** denotes the class of promise problems possessing interactive proof systems.
- **HV-SZKP** and **HV-CZKP** denote the classes of promise problems having honest-verifier statistical and computational zero-knowledge proofs, respectively. Analogously, **HV-SZKA** and **HV-CZKA** denote the classes of promise problems having 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 versus 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 than 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*: In interactive proofs and interactive arguments, and in their zero-knowledge analogues, we allow the honest prover to be computationally unbounded, unless we specify *efficient prover*, which means a polynomial-time honest prover strategy given a witness for membership. It was shown in [NV] that 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 $\text{view}_A(A(a), B(b))(x)$ 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 $\text{view}_B(A(a), B(b))(x)$ 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 $\langle P, V\rangle(x) = \text{view}_V(P, V)(x)$. 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* [BM].)

Definition 2.10 (interactive proofs)**.** An interactive protocol (P, V) is an *interactive proof system* for a promise problem Π if exist functions $c, s : \mathbb{N} \to [0, 1]$ such that $1 - c(n) > s(n) + 1/\text{poly}(n)$ and the following conditions hold.

- *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 [GS, FGM+].

Definition 2.11 (interactive arguments)**.** We say that (P, V) is an *interactive argument system* for Π if the soundness condition in Definition 2.10 holds against all nonuniform PPT P^*, instead of every (computationally unbounded) P^*. Specifically, we require both the efficiency and completeness conditions in Definition 2.10 to hold, and the new (weaker) soundness condition is the following.

- *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.[6]

Definition 2.12 (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}.

[6] 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 2.13 (auxiliary-input zero knowledge)**.** [7] 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

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

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. To avoid cumbersome terminologies, we often drop the prefix auxiliary input and just use *zero knowledge* to actually mean auxiliary-input zero knowledge.

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 [GO]. Plain zero knowledge (i.e., without auxiliary inputs) is not closed under sequential composition [GK1], 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 2.14 (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.

[7] Our formulation of auxiliary-input zero knowledge is slightly different than, but equivalent to, the definition in the textbook [Gol2]. 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 [Gol2, 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 2.6.

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 [Bar] 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 [GK1]. Nevertheless, our results will show that, when ignoring round 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**.

Recall that **NP**, informally stated, is the class of problems that can be verified in polynomial time given a valid "witness." To formally define the relationship between an instance and its corresponding valid witnesses, we consider a relation W and say that W is *polynomial time* if deciding whether an element is in W can be done in polynomial time in the length of the first component of the input (this is typically the length of the problem instance). With this, a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathbf{NP}$ if there exist a polynomial-time binary relation $W \subseteq \{0,1\}^* \times \{0,1\}^*$ such that the following two conditions hold:

- for every $x \in \Pi_{\mathrm{Y}}$, there exists a w with $(x,w) \in W$;
- for every $x \in \Pi_{\mathrm{Y}}$, and for every w, it is the case that $(x,w) \notin W$.

Any polynomial-time binary relation that satisfies the above two conditions is said to be an **NP**-*relation* for the problem Π.

For **MA**, the probabilistic analog of **NP**, we generalize the relation W to allow for randomness; specifically, we expand the domain of $W \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$. To relate it with the **NP** case above, we abuse notation and write $(x,w) \in W$ if $\Pr_r[(x,w,r) \in W] \geq 2/3$, and write $(x,w) \notin W$ if $\Pr_r[(x,w,r) \in W] \leq 1/3$. Then, a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathbf{MA}$ if there exist a polynomial-time relation $W \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ such that the following two conditions hold:

- for every $x \in \Pi_{\mathrm{Y}}$, there exists a w with $(x,w) \in W$, namely $\Pr_r[(x,w,r) \in W] \geq 2/3$;
- for every $x \in \Pi_{\mathrm{Y}}$, and for every w, it is the case that $(x,w) \notin W$, namely $\Pr_r[(x,w,r) \in W] \leq 1/3$.

Any polynomial-time relation that satisfies the above two conditions is said to be an **MA**-*relation* for the problem 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 its strategy on problem instance x is computable in polynomial time given w as auxiliary input, where $(x,w) \in W$ and W is an **NP**-relation [resp., **MA**-relation] for Π. We note that an equivalent formulation of **MA** is the class of problems having efficient-prover argument systems (cf., [BLV]). This means efficient provers is defined, without loss of generality, only for problems in **MA**.

Remarks on the definitions. Our definitions mostly follow the now-standard definitions of zero-knowledge proofs as presented in [Gol2], 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 [NV] 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., [GK2, SV]), we extend all of the definitions to promise problems $\Pi = (\Pi_Y, \Pi_N)$ 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 [Gol1] and [BLV, 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 versus strict polynomial-time simulators. (This equivalence is achieved following a similar line of reasoning as [Vad].)

3 Unconditional Characterizations of Zero Knowledge

In this section, we provide *unconditional* characterizations of zero knowledge that would among other things allow us to establish our Symmetry Theorem between computational zero knowledge and computational soundness (Theorem 1.2). We first present our main characterization theorems in Section 3.1, which expands upon Theorem 1.4. The steps involved in proving these characterization theorems are outlined in Section 3.2, and lemmas needed to establish these theorems are given in Sections 3.3, 3.4, and 3.5.

3.1 Our Main Characterization Theorems

In this subsection, we elaborate upon the SZKP–OWF Characterization of Zero Knowledge Theorem (Theorem 1.4). Specifically, we state four theorems giving a variety of equivalent characterizations of the classes **SZKP**, **CZKP**, **CZKA**, and **SZKA**. The ones for zero-knowledge arguments, namely **CZKA** and **SZKA**, are new; the other for zero-knowledge proofs, namely **CZKP** and

SZKP, contain results 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 satisfying the SZKP–OWF Condition, 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, and do not necessarily preserve other efficiency measures like round complexity, unless explicitly mentioned.

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

Theorem 3.1 (**SZKP** Characterization Theorem [Oka, GSV, NV, HORV])**.** *For every problem* $\Pi \in \mathbf{IP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{HV\text{-}SZKP}$.
2. Π *satisfies the* SZKP–OWF Condition *without* owf *instances.*
3. Π *has an instance-dependent commitment scheme that is statistically hiding on the* yes *instances and statistically binding on the* no *instances. Moreover, the scheme is public coin.*
4. $\Pi \in \mathbf{SZKP}$*, and the statistical zero-knowledge proof system for* Π *has a black-box simulator, is public coin, and has perfect completeness. Furthermore, if* $\Pi \in \mathbf{NP}$*, the proof system has an efficient prover.*

Theorem 3.2 (**CZKP** Characterization Theorem [Vad, NV, HORV])**.** *For every problem* $\Pi \in \mathbf{IP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{HV\text{-}CZKP}$.
2. Π *satisfies the* SZKP–OWF Condition *without* owf no *instances.*
3. Π *has an instance-dependent commitment scheme that is computationally hiding on the* yes *instances and statistically binding on the* no *instances. Moreover, the scheme is public coin.*
4. $\Pi \in \mathbf{CZKP}$*, and the computational zero-knowledge proof system for* Π *has a black-box simulator, is public coin, and has perfect completeness. Furthermore, if* $\Pi \in \mathbf{NP}$*, the proof system has an efficient prover.*

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

Theorem 3.3 (**SZKA** Characterization Theorem)**.** *For every problem* $\Pi \in \mathbf{NP}$*, the following conditions are equivalent.*

1. $\Pi \in \mathbf{HV\text{-}SZKA}$.
2. Π *satisfies the* SZKP–OWF Condition *without* owf yes *instances.*
3. Π *has an instance-dependent commitment scheme that is statistically hiding on the* yes *instances and computationally binding on the* no *instances. Moreover, the scheme is public coin.*

4. $\Pi \in$ **SZKA**, *and the statistical zero-knowledge argument system for* Π *has a black-box simulator, is public coin, has perfect completeness, and an efficient prover.*

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

1. $\Pi \in$ **HV-CZKA**.
2. Π *satisfies the* SZKP–OWF Condition.
3. Π *has an instance-dependent commitment scheme that is computationally hiding on the* YES *instances and computationally binding on the* NO *instances. Moreover, the scheme is public coin.*
4. $\Pi \in$ **CZKA**, *and the computational zero-knowledge proof system for* Π *has a black-box simulator, is public coin, has perfect completeness, and an efficient prover.*

We prove Theorems 3.3 and 3.4 using lemmas established in Sections 3.3, 3.4, and 3.5. Notice that in these theorems involving zero knowledge arguments, we have restricted Π to be in **NP** in contrast to the theorems involving zero-knowledge proofs (Theorems 3.1 and 3.2), which are naturally restricted 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). 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 3.1 to 3.3 above).

3.2 Steps of Our Proof

Having stated our main characterization theorems in the previous subsection, we now provide an outline of the steps involved in establishing these characterization theorems.

1. We show that every problem Π possessing a (honest-verifier) zero-knowledge protocol satisfies the SZKP–OWF Condition. Depending on the zero knowledge and soundness guarantee, the types of SZKP–OWF Condition that Π satisfies will differ (in whether the sets of OWF YES instances and OWF NO instances are empty or nonempty). This extends the unconditional characterization work of [Vad] for zero-knowledge proof systems to the more general zero-knowledge argument systems, and is in Section 3.3.

2. Next, we show that every problem Π satisfying the SZKP–OWF Condition yields an *instance-dependent commitment scheme* for Π. This is based on the techniques of [NOV, NV, HR, HORV], and is in Section 3.4.

3. Finally, we show that every problem $\Pi \in$ **NP** having instance-dependent commitments allow us to construct zero-knowledge argument systems for Π with desirable properties like perfect completeness, black-box zero knowledge, public coins, and an efficient prover. This is done by substituting instance-dependent commitments for standard (non-instance-dependent) commitments used in existing zero-knowledge protocols like the Goldreich–Micali–Wigderson [GMW] zero-knowledge protocol for **NP**, and is in Section 3.5.

A summary of the steps involved in establishing our characterization theorems, together with their corresponding lemmas, is given in Figure 1.

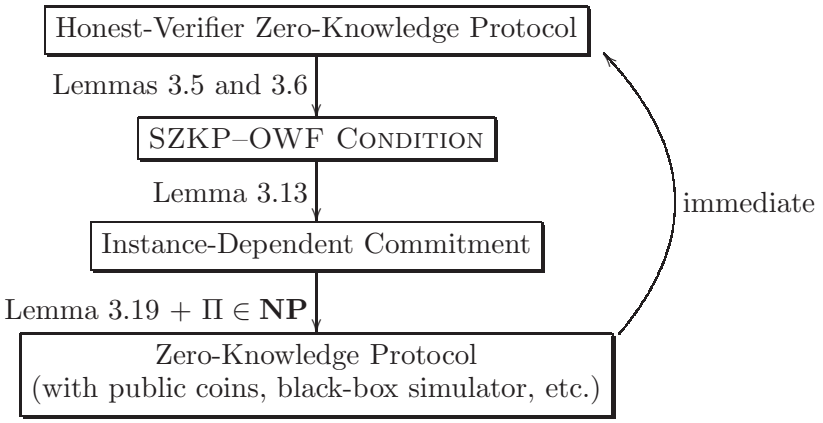

Figure 1: Steps of our proof.

3.3 From Zero-Knowledge Protocols to SZKP–OWF Characterizations

In this subsection, we show that problems possessing (honest verifier) zero-knowledge arguments satisfy the SZKP–OWF Condition. Specifically, we prove that for every problem Π having a zero-knowledge argument also satisfies the SZKP–OWF Condition. This involving establishing a set of instances $I \subseteq \Pi_Y \cup \Pi_N$ such that $(\Pi_Y \setminus I, \Pi_N \setminus I) \in \mathbf{SZKP}$, and from which instance-dependent one-way functions can be constructed. The main difference from [Vad] is that [Vad] characterizes only zero-knowledge proofs and has no owf no instances, namely $I \cap \Pi_N = \emptyset$. In other words, the characterizations of [Vad] satisfy the SZKP–OWF Condition without owf no instances.

We state a lemma establishing SZKP–OWF Characterizations for zero-knowledge *proofs*. This lemma follows from the works of [Oka, GSV, Vad], but is given for comparison.

Lemma 3.5 ([Oka, GSV, Vad])**.** *If problem* $\Pi \in$ **HV-CZKP**, *then* Π *satisfies the* SZKP–OWF Condition *without* owf no *instances, namely* $I \cap \Pi_N = \emptyset$. *In addition, if* $\Pi \in$ **HV-SZKP**, *then* Π *satisfies the* SZKP–OWF Condition *without* owf *instances, namely* $I = \emptyset$.

Next, we give analogous SZKP–OWF Characterizations for zero-knowledge *arguments*.

Lemma 3.6. *If problem* $\Pi \in$ **HV-CZKA**, *then* Π *satisfies the* SZKP–OWF Condition. *In addition, if* $\Pi \in$ **HV-SZKA**, *then* Π *satisfies the* SZKP–OWF Condition *without* owf yes *instances, namely* $I \cap \Pi_Y = \emptyset$.

Proof Idea. Proving that $\Pi \in$ **HV-CZKA** satisfies the SZKP–OWF Condition involves establishing a set I with an instance-dependent one-way on I and $(\Pi_Y \setminus I, \Pi_N \setminus I) \in \mathbf{SZKP}$. To do so, we provide a separate analysis for the yes and no instances; namely, we show that there exist sets $I_Y \subseteq \Pi_Y$ and $I_N \subseteq \Pi_N$ such that instance-dependent one-way functions can be constructed on these sets, and that $(\Pi_Y \setminus I_Y, \Pi_N \setminus I_N) \in \mathbf{SZKP}$. These instance-dependent one-way functions

f_x and g_x on I_Y and I_N, respectively, can be combined into a single instance-dependent one-way function on $I \stackrel{\text{def}}{=} I_Y \cup I_N$ by concatenating the functions f_x and g_x.

The sets I_Y and I_N are defined based on the simulator S for the zero-knowledge protocol of $\Pi \in$ **HV-CZKA**. Following Fortnow [For], we consider a *simulation-based prover* P_S and corresponding *simulation-based verifier* V_S. Informally, 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. 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 $x \in \Pi_{\text{Y}}$.
3. By assuming, without loss of generality, that the simulator always outputs accepting transcripts, it holds that P_S makes V_S accepts with probability 1, on every x.

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 below), we define sets I_Y and I_N as follows:

- I_Y contains instances $x \in \Pi_{\text{Y}}$ for which V_S is *statistically different* from V, and
- I_N contains instances $x \in \Pi_{\text{N}}$ for which V_S is *statistically similar* to V.

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

1. On I_Y, 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_Y, as shown in [Vad].
2. On I_N, 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 the computational soundness of (P, V), it must be the case that P_S is not PPT. Using techniques from Ostrovsky [Ost], this allows us to convert the simulator S into an instance-dependent distributional one-way function g_x.[8] Then by Proposition 2.4, due to Impagliazzo and Luby [IL], we can obtain an instance-dependent one-way function from g_x.
3. To see that $(\Pi_{\text{Y}} \setminus I_Y, \Pi_{\text{N}} \setminus I_N) \in$ **SZKP**, we observe the following: for those YES instances not in I_Y—that is, instances in $\Pi_{\text{Y}} \setminus I_Y$—the simulated verifier V_S is statistically similar to V. And for those NO instances not in I_N—that is, instances in $\Pi_{\text{N}} \setminus I_N$—the simulated verifier V_S is statistically different from V. This gap in the statistical properties allows us to reduce promise problem $(\Pi_{\text{Y}} \setminus I_Y, \Pi_{\text{N}} \setminus I_N)$ to one of the complete problems for **SZKP** [SV, GV, Vad].

[8] If g_x is not distributionally one-way, then P_S can be made to be efficient, hence contradicting the computational soundness of (P, V). Interestingly, Ostrovsky [Ost] uses the assumption that g_x 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 [Ost], we instead invert S on the NO instances to contradict the computational soundness of (P, V).

Proof of Lemma 3.6. Let (P, V) be a zero-knowledge argument system for Π, with simulator S. We now proceed as in the proof of [Vad] and modify our interactive protocol (P, V) to satisfy the following (standard) additional properties.

- 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 $r(|x|)$ random coin tosses in the last message. (Without loss of generality, we may assume that $r(|x|) \geq |x|$.) 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 S always outputs *accepting transcripts*, where we call a sequence τ 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 τ, we denote by τ_i the *prefix* of τ consisting of the first i messages. For readability, we often drop the input x from the notation, for instance using $\ell = \ell(|x|)$, $\langle P, V\rangle = \langle P, V\rangle(x)$, $r = r(|x|)$, and so forth. 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.

Using the simulator S, we define the simulation-based prover P_S as follows: On input x and execution prefix τ_{2i}, for $i = 1, 2, \ldots, \ell - 1$, do the following:

1. If simulator $S(x)$ outputs a transcript that begins with τ_{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 α with probability $p_\alpha = \Pr\left[S(x)_{2i} = \tau_{2i-1} \circ \alpha | S(x)_{2i-1} = \tau_{2i-1}\right]$.

The simulation-based verifier V_S can be defined analogously as follows: On input x and execution prefix τ_{2i-1}, for $i = 1, 2, \ldots, \ell$, do the following:

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

Observe that $\langle P_S, V_S\rangle(x)$ is identically distributed to $S(x)$, for every x. Following [AH, PT, GV, Vad], we consider the following quantity:

$$h(x) = \sum_{i=1}^{\ell} \left[\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P_S, V_S\rangle(x)_{2i}) - \mathrm{H}(\langle P_S, V_S\rangle(x)_{2i-1})\right] \quad , \qquad (1)$$

where $\mathrm{H}(\cdot)$ denotes the *(Shannon) entropy* measure, which is given by $\mathrm{H}(X) = \mathrm{E}_{x \leftarrow X}\left[\log(1/\Pr[X = x])\right]$.

From [AH, PT, GV], we know that for every $x \in \{0,1\}^*$, and every prover strategy P',

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

The above sum in (2) measures the total entropy contributed by the honest verifier's messages, and hence it is natural that this should equal $r(|x|)$, the number of coin tosses of the honest verifier. This is because the honest verifier reveals all its coin tosses at the end.

From (1) and (2), we observe that how close the value of $h(x)$ gets to $r(|x|)$ is a measure of how close the simulation-based verifier V_S is from the honest verifier V (when interacting with P_S). Following our intuition in the proof sketch above, we let I_Y be the set of instances $x \in \Pi_{\mathrm{Y}}$ for which the V_S is "far" from the honest verifier V, and we let I_N be the set of instances $x \in \Pi_{\mathrm{N}}$ for which the V_S is "close" to V. Formally, we define:

$$I_Y = \{x \in \Pi_{\mathrm{Y}} : h(x) < r(|x|) - 1/q(|x|)\} \ ;$$
$$I_N = \{x \in \Pi_{\mathrm{N}} : h(x) > r(|x|) - 2/q(|x|)\} \ ,$$

where the polynomial $q(|x|) = 256 \cdot \ell(|x|)$.

Having defined sets I_Y and I_N, Lemma 3.6 is established by the following claims. The first three are proven in the same way as in [Vad], and hence we defer their proofs to Appendix A.

Claim 3.7. Problem $(\Pi_{\mathrm{Y}} \setminus I_Y, \Pi_{\mathrm{N}} \setminus I_N) \in \mathbf{SZKP}$.

Claim 3.8. There exists an instance-dependent one-way function on I_Y.

Claim 3.9. For $\Pi \in \mathbf{HV\text{-}SZKA}$, we can take $I_Y = \emptyset$.

The main novelty in our analysis is the following claim.

Claim 3.10. There exists an instance-dependent one-way function on I_N.

Proof of Claim. To get an instance-dependent one-way function on I_N, we use the following idea of Ostrovsky [Ost]: 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 the function g_x, whose purpose is to output messages of the simulator, as follows:

$$g_x(i, \omega) = (x, i, S(x; \omega)_{2i}) \ . \tag{3}$$

Note that g_x is polynomial-time computable because the simulator S runs in polynomial time. If g_x is *not* distributionally one-way (in the sense of Definition 2.3), 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 τ_{2i} after the verifier has responded in round $2i$, into the inversion algorithm of g_x to obtain the simulation-based prover response for round $2i+1$. In doing so, we contradict the computational soundness property of (P, V). This argument is captured by following proposition, whose proof is given in Appendix A.

Proposition 3.11 (based on [Ost, Lemma 1]). [9] *Let g_x be as in* (3)*. For every set $K \subseteq \{0,1\}^*$, if g_x is* not *an instance-dependent distributionally one-way function on K, then for every polynomial p, there exists a nonuniform PPT prover $\widetilde{P}$ such that*

$$\Delta(\langle \widetilde{P}, V\rangle(x), S(x)) \leq \ell(|x|) \cdot \left(\frac{1}{p(|x|)} + 2 \cdot \Delta(\langle P_S, V\rangle(x), S(x))\right) \ ,$$

for infinitely many $x \in K$.

This leaves us to upper bound $\Delta(\langle P_S, V\rangle, S)$ in order to obtain an upper bound on $\Delta(\langle \widetilde{P}, V\rangle, S)$, and hence contradict the computational soundness of V (because S always outputs accepting transcripts). Recall that for every $x \in I_N$, we have $h > r - 2/q$. From [AH, PT, GV], we know that $h = r - \mathrm{KL}(\langle P_S, V\rangle, S)$, where KL is the *Kullback-Leibler* distance defined as $\mathrm{KL}(X,Y) = \mathrm{E}_{\alpha \leftarrow X}\big[\log(\Pr[X = \alpha]) - \log(\Pr[Y = \alpha])\big]$. (See [GV, Lemma 2.2].) Hence, we get $\mathrm{KL}(\langle P_S, V\rangle, S) < 2/q$. Using the fact that for any random variables X and Y, $\mathrm{KL}(X,Y) \geq (1/2) \cdot (\Delta(X,Y))^2$ [CT, Lemma 12.6.1], we get that for all $x \in I_N$,

$$\Delta(\langle P_S, V\rangle, S) < 2/\sqrt{q} = 1/(8 \cdot \ell) \ , \tag{4}$$

since $q = 256 \cdot \ell$.

Now by Proposition 3.11, if g_x is not distributionally one-way on I_N, we can take $I_N = K$ and choose $p(|x|) = 4 \cdot \ell(|x|)$, to get a nonuniform PPT $\widetilde{P}$ such that

$$\begin{aligned} \Delta(\langle \widetilde{P}, V\rangle, S) &\leq \ell \cdot (1/p + 2 \cdot \Delta(\langle P_S, V\rangle, S)) \\ &= 1/4 + 2 \cdot \ell \cdot \Delta(\langle P_S, V\rangle, S) \\ &< 1/2 \ . \qquad \text{(by (4))} \end{aligned}$$

And since the simulator S always produce accepting transcripts, we have

$$\Pr[(\widetilde{P}, V)(x) = \texttt{accept}] \geq 1/2 \ ,$$

for infinitely many $x \in I_N$. This contradicts the computational soundness of (P,V). Therefore, g_x must be a distributionally one-way function on I_N. By Proposition 2.4 (due to Impagliazzo and Luby [IL]), g_x can be converted into an instance-dependent (standard) one-way function on I_N, as desired. □

Let us see how the above five claims establish Lemma 3.6. Define set $I = I_Y \cup I_N$. This means that the promise problem $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) = (\Pi_{\mathrm{Y}} \setminus I_Y, \Pi_{\mathrm{N}} \setminus I_N)$, and Claim 3.7 places this problem in **SZKP**. Claims 3.8 and 3.10 give us instance-dependent one-way functions on I_Y and I_N, respectively; to obtain a single instance-dependent one-way function on $I = I_Y \cup I_N$, we use the following claim.

Claim 3.12. For any sets $J, K \subseteq \{0,1\}^*$, if there exist instance-dependent one-way functions on J and there exist instance-dependent one-way functions on K, then there exist instance-dependent one-way functions on $J \cup K$.

[9] As pointed out to us by Lilach Bien, the statement and application of this proposition in the original version of our paper [OV, Lemma 4.8] erroneously neglected the dependence on $\Delta(\langle P_S, V\rangle(x), S(x))$.

Proof of Claim. Let f_x and g_x be any instance-dependent one-way function on J and K, respectively. Then, $h_x(y,z) = (f_x(y), g_x(z))$ is an instance-dependent one-way function on $J \cup K$. This is because inverting h_x involves inverting both f_x and g_x, at least one of which is hard to invert on $J \cup K$. □

Therefore, by Claim 3.12 above, we know that $\Pi \in$ **HV-CZKA** satisfies the SZKP–OWF CONDITION. Furthermore, if $\Pi \in$ **HV-SZKA**, Claim 3.9 tells us that $I_Y = \emptyset$, and hence $I \cap \Pi_Y = I_Y = \emptyset$, giving us that Π satisfies the SZKP–OWF CONDITION without OWF YES instances. □

3.4 From SZKP–OWF Characterization to Instance-Dependent Commitment Schemes

In this subsection, we show that every problem Π satisfying the SZKP–OWF CONDITION yields an instance-dependent commitment scheme for Π. This is obtained by combining statistically-binding commitments from one-way functions [Nao, HILL], statistically-hiding commitments from one-way functions [NOV, HR], and instance-dependent commitments for **SZKP** [NV, HORV]. In the original version of this paper [OV], our instance-dependent commitment scheme inherited a certain "1-out-of-2" binding property from [NV, NOV]. This property is weaker and more complicated than the standard binding property of commitments, but sufficed for establishing our main theorems (Theorems 1.2 and 1.4). Due to improvements by [HR, HORV], it is now possible to construct instance-dependent commitments with the standard binding property, and hence we use standard-binding commitments to simplify our presentation.

Lemma 3.13. *The following conditions hold for problems* Π *satisfying the* SZKP–OWF CONDITION.

- *If* Π *satisfies the* SZKP–OWF CONDITION *without* OWF NO *instances [resp., without* OWF *instances], then it has an instance-dependent commitment scheme that is computationally [resp., statistically] hiding on the* YES *instances and statistically binding on the* NO *instances.*
- *If* Π *satisfies the* SZKP–OWF CONDITION *[resp., without* OWF YES *instances], then it has an instance-dependent commitment scheme that is computationally [resp., statistically] hiding on the* YES *instances and computationally binding on the* NO *instances.*

Furthermore, all the above instance-dependent commitment schemes are public coin.

The proof of Lemma 3.13, tying together all the following propositions and claims, is given at the end of this subsection. Before stating our propositions and claims, we provide an outline of what we intend to construct in the next paragraph.

Given that problem Π satisfies the SZKP–OWF CONDITION, we let the set of OWF YES instances be denoted as $I_Y = I \cap \Pi_{\mathrm{Y}}$, and the set of OWF NO instances be denoted as $I_N = I \cap \Pi_{\mathrm{N}}$. Our task of constructing an instance-dependent commitment scheme for Π is broken into following four steps: (1) construct an instance-dependent commitment scheme for the problem $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) \in$ **SZKP**, (2) construct an instance-dependent commitment scheme for the problem $(I_Y, \overline{I_Y})$, (3) construct an instance-dependent commitment scheme for the problem $(\overline{I_N}, I_N)$, and (4) combine all these three instance-dependent commitment schemes into a single instance-dependent commitment scheme for Π. We will explain why these four steps yield an instance-dependent commitment scheme for Π in the proof of Lemma 3.13, given at the end of this subsection.

Step 1: The instance-dependent commitment for the problem $(\Pi_Y \setminus I_Y, \Pi_N \setminus I_N) \in \mathbf{SZKP}$ follows from [HORV] (which builds on [NV]).

Proposition 3.14 ([HORV])**.** *For any problem $\Gamma \in \mathbf{SZKP}$, problem Γ has an instance-dependent commitment scheme that is statistically hiding on the* YES *instances and statistically binding on the* NO *instances. Moreover, the instance-dependent commitment scheme obtained is public coin.*

Step 2: Notice that the instance-dependent commitments given by the above proposition do not guarantee hiding or binding properties on the OWF instances sets I_Y and I_N. Nevertheless, we noted in [Vad], we can use the instance-dependent one-way functions on I_Y to construct instance-dependent commitment schemes that are computationally hiding on I_Y and statistically binding elsewhere, based on Naor's [Nao] commitment scheme. This is because Naor's scheme can be based on any one-way function [HILL], and the statistical binding property of the scheme does not depend on the one-way security of the function.

Proposition 3.15 (based on [Nao, HILL])**.** *For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(K, \overline{K})$ has an instance-dependent commitment scheme that is computationally hiding on the* YES *instances (namely, instances in K), and statistically binding on the* NO *instances (namely, instances in $\overline{K}$). Moreover, the instance-dependent commitment scheme obtained is public coin.*

Step 3: We construct instance-dependent commitment schemes that are computationally binding on I_N and statistically hiding elsewhere, based on the fact that statistically hiding and computationally binding commitments can be constructed from any one-way function [NOV, HR].

Proposition 3.16 (based on [NOV, HR])**.** *For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(\overline{K}, K)$ has an instance-dependent commitment that is statistically hiding on the* YES *instances (namely, instances in $\overline{K}$), and computationally binding on the* NO *instances (namely, instances in K). Moreover, the instance-dependent commitment scheme obtained is public coin.*

Step 4: Finally, we use standard methods to combine the three instance-dependent commitment schemes that we have constructed into a single instance-dependent commitment scheme for Π. The first method gives a combined scheme for the intersection of two problems.

Claim 3.17. Suppose problems $\Gamma = (\Gamma_Y, \Gamma_N)$ and $\Gamma' = (\Gamma'_Y, \Gamma'_N)$ have instance-dependent commitment schemes Com_x and Com'_x, respectively. Then problem $\Gamma \cap \Gamma' = (\Gamma_Y \cap \Gamma'_Y, \Gamma_N \cup \Gamma'_N)$ has an instance-dependent commitment scheme Com''_x with the following properties.

- Com''_x is statistically [resp., computationally] hiding if both Com_x and Com'_x are statistically [resp., computationally] hiding.
- Com''_x is statistically [resp., computationally] binding if either of Com_x or Com'_x is statistically [resp., computationally] binding.
- Com''_x is public coin if both Com_x and Com'_x are public coin.

Proof. In commitment scheme Com''_x, the sender commits to b by committing to b in both schemes Com_x and Com'_x, with the execution of both schemes done in parallel. The claimed properties of Com''_x follow by inspection. □

The second method provides a combined scheme for the union of two problems.

Claim 3.18. Suppose problems $\Gamma = (\Gamma_{\mathrm{Y}}, \Gamma_{\mathrm{N}})$ and $\Gamma' = (\Gamma'_{\mathrm{Y}}, \Gamma'_{\mathrm{N}})$ have instance-dependent commitment schemes Com_x and Com'_x, respectively. Then problem $\Gamma \cup \Gamma' = (\Gamma_{\mathrm{Y}} \cup \Gamma'_{\mathrm{Y}}, \Gamma_{\mathrm{N}} \cap \Gamma'_{\mathrm{N}})$ has an instance-dependent commitment scheme Com''_x with the following properties.

- Com''_x is statistically [resp., computationally] hiding if either of Com_x or Com'_x is statistically [resp., computationally] hiding.
- Com''_x is statistically [resp., computationally] binding if both Com_x and Com'_x are statistically [resp., computationally] binding.
- Com''_x is public coin if both Com_x and Com'_x are public coin.

Proof. In commitment scheme Com''_x, the sender on input bit b, first secret shares b into two shares, b_1 and b_2, with the property that $b_1 \oplus b_2 = b$ and each b_i is uniform in $\{0,1\}$. (This can be done by choosing a random $b_1 \leftarrow \{0,1\}$, and setting $b_2 = b_1 \oplus b$.) The sender then commits to b by committing to bits b_1 and b_2 in schemes Com_x and Com'_x, respectively. The execution of schemes Com_x and Com'_x is done in parallel.

The hiding property follows from the fact that bit b remains hidden as long as one of the bits b_1 or b_2 remains hidden. Then binding property follows from the fact that $b = b_1 \oplus b_2$, and hence b is bounded to a fixed value if both b_1 and b_2 are bounded to fixed values. The public coin property and round complexity of Com''_x follow by inspection. □

Having established the propositions and claims that we need, we now prove Lemma 3.13.

Proof of Lemma 3.13. Given that problem Π satisfies the SZKP–OWF Condition, let I be the set of OWF instances, and let the OWF YES instances be $I_Y = I \cap \Pi_{\mathrm{Y}}$ and the OWF NO instances be $I_N = I \cap \Pi_{\mathrm{N}}$. By Propositions 3.14, 3.15, and 3.16, we have three instance-dependent commitment schemes, call them $\mathsf{Com}^{(1)}_x$, $\mathsf{Com}^{(2)}_x$, and $\mathsf{Com}^{(3)}_x$, for the problems $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) \in \mathbf{SZKP}$, $(I_Y, \overline{I_Y})$, and $(\overline{I_N}, I_N)$, respectively. Moreover, all three schemes are public coin.

If Π satisfies the SZKP–OWF Condition without OWF instances, then set $I = \emptyset$, and hence $\mathsf{Com}^{(1)}_x$ suffices to be our instance-dependent commitment scheme for Π. If Π satisfies the SZKP–OWF Condition without OWF NO instances, then $I_N = I \cap \Pi_{\mathrm{N}} = \emptyset$. Consequently, we do not need scheme $\mathsf{Com}^{(3)}_x$, and can just combine schemes $\mathsf{Com}^{(1)}_x$ and $\mathsf{Com}^{(2)}_x$ in a manner prescribed by Claim 3.18 to get an instance-dependent commitment scheme for Π.

Analogously, if Π satisfies the SZKP–OWF Condition without OWF YES instances, then $I_Y = I \cap \Pi_{\mathrm{Y}} = \emptyset$. Consequently, we do not need scheme $\mathsf{Com}^{(2)}_x$, and can just combine schemes $\mathsf{Com}^{(1)}_x$ and $\mathsf{Com}^{(3)}_x$ in a manner prescribed by Claim 3.17 to get an instance-dependent commitment scheme for Π. Finally, if Π satisfies the SZKP–OWF Condition, we first combine schemes $\mathsf{Com}^{(1)}_x$ and $\mathsf{Com}^{(2)}_x$ in a manner prescribed by Claim 3.18 to get an instance-dependent commitment scheme for $(\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus I_N)$, and then combine this scheme with $\mathsf{Com}^{(3)}_x$ in a manner prescribed by Claim 3.17 to get an instance-dependent commitment scheme for Π.

The hiding, binding, and public coin properties of the instance-dependent commitment scheme for Π follow by inspection. □

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

Having obtained instance-dependent commitments in the previous subsection, we now use these commitments to construct unconditional zero-knowledge protocols for problems $\Pi \in \mathbf{NP}$ having these instance-dependent commitments. We observe that the existing zero-knowledge protocols for **NP** require complexity assumptions because they use standard (non-instance dependent) commitments, and standard commitments are not known to exist unconditionally. Therefore, as observed in [IOS], we can remove the complexity assumptions needed by substituting standard commitments for instance-dependent commitments in these existing protocols. Specifically, we do this substitution in the Goldreich–Micali–Wigderson [GMW] zero-knowledge protocol for **NP**.

Lemma 3.19 (based on [GMW])**.** *If problem $\Pi \in \mathbf{NP}$ has an instance-dependent commitment scheme Com_x, then it has a zero-knowledge protocol (P, V) with the following properties.*

- *(P, V) is statistical [resp., computational] zero knowledge if Com_x is statistically [resp., computationally] hiding on the* YES *instances. Moreover, (P, V) has a black-box simulator.*
- *(P, V) is a proof [resp., argument] system if Com_x is statistically [resp., computationally] binding on the* NO *instances.*
- *(P, V) has perfect completeness and has an efficient prover.*
- *(P, V) is public coin if Com_x is public coin.*

3.6 Putting It All Together

We now show how our lemmas in Sections 3.3, 3.4, and 3.5 imply our main characterization theorems in Section 3.1.

Proof of Theorems 3.3 and 3.4. The implications for both theorems are captured by the same lemmas, so we can conveniently state them together.

(1) ⇒ (2) is established by Lemma 3.6.

(2) ⇒ (3) is established by Lemma 3.13.

(3) ⇒ (4) is established by Lemma 3.19. This is the only step that requires the problem $\Pi \in \mathbf{NP}$.

(4) ⇒ (1) follows directly from definition. □

Acknowledgements

We are grateful to Lilach Bien for pointing out an error in the statement and use of Lemma 4.8 in the original version of our paper [OV], which is corrected in Proposition 3.11. We thank Oded Goldreich and the anonymous *EUROCRYPT 2007* reviewers for their helpful comments. We also thank Iftach Haitner and Omer Reingold for allowing us to simplify the presentation of our results with their wonderful result [HR] and the follow-up [HORV].

A Proofs from Section 3.3

We restate and prove Claims 3.7, 3.8, and 3.9, and Proposition 3.11 from Section 3.3. The three claims are proven using techniques from [Vad], and Proposition 3.11 is based on ideas from Ostrovsky [Ost]. Recall that (P, V) is the zero-knowledge argument system for Π, with simulator S.

Before proving the above claims and proposition, 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

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

Next, recall the definition of $h(x)$ as stated by (1) in Section 3.3:

$$h(x) = \sum_{i=1}^{\ell} \left[\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \mathrm{H}(S(x)_{2i}|S(x)_{2i-1}) \ , \tag{5}$$

where $\mathrm{H}(\cdot)$ denotes the *(Shannon) entropy* measure, which is given by $\mathrm{H}(X) = \mathrm{E}_{x\leftarrow X}\left[\log(1/\Pr[X=x])\right]$. The second equality in (5) follows the fact that the output of S_{2i} contains S_{2i-1}, and hence $\mathrm{H}(S_{2i}, S_{2i-1}) = \mathrm{H}(S_{2i})$.

Finally, recall that from (2) in Section 3.3, we have that for every $x \in \{0,1\}^*$, and every prover strategy P', the number of coins used by the honest verifier, denoted by $r(|x|)$, is:

$$r(|x|) = \sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P', V\rangle(x)_{2i}) - \mathrm{H}(\langle P', V\rangle(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \mathrm{H}(\langle P', V\rangle(x)_{2i}|\langle P', V\rangle(x)_{2i-1}) \ , \tag{6}$$

with the second equality following from the fact that the output of $\langle P', V\rangle_{2i}$ contains $\langle P', V\rangle_{2i-1}$, and hence $\mathrm{H}(\langle P', V\rangle_{2i}, \langle P', V\rangle_{2i-1}) = \mathrm{H}(\langle P', V\rangle_{2i})$.

Restatement of Claim 3.7. *Problem* $(\Pi_{\mathrm{Y}} \setminus I_Y, \Pi_{\mathrm{N}} \setminus I_N) \in \mathbf{SZKP}$.

Proof. We note the following proposition.

> **Proposition A.1** (based on [Vad, Proposition 3.2])**.** 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**.

Given the above proposition, it suffices to show a reduction from $(\Pi_{\mathrm{Y}} \setminus I_Y, \Pi_{\mathrm{N}} \setminus I_N)$ to Conditional Entropy Approximation. Our reduction is as follows: On input x, we construct circuits X and Y that sample from the following (joint) random variables.

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

When $x \in \Pi_{\mathrm{Y}} \setminus I_Y$, we have $h(x) > r - 1/q$, and hence:

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) = \frac{h}{\ell} > \frac{r-1/q}{\ell} = \frac{r}{\ell} - \frac{1}{q\cdot\ell} \ .$$

And when $x \in \Pi_{\mathrm{N}} \setminus I_N$, we have have $h(x) < r - 2/q$, and hence:

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

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

Restatement of Claim 3.8. *There exists an instance-dependent one-way function on I_Y.*

Proof. We note the following proposition.

> **Proposition A.2** (based on [Vad, Lemma 3.10])**.** Let $K \subseteq \{0,1\}^*$ be any set. Assume that there exists a polynomial-time computable mapping that maps every $x \in K$ to samplable joint distributions (X,Y) and a parameter r such that $\mathrm{H}(X|Y) \le r-1$, but $\mathrm{H}(X'|Y') \ge r$ for some (X',Y') indistinguishable from (X,Y). Then there exists an instance-dependent one-way function on K.

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 3.7 above, is indistinguishable from $(X',Y') = (\langle P,V\rangle_{2L}, \langle P,V\rangle_{2L-1})$, where random variable L denotes an independent uniform random element of $\{1,\ldots,\ell\}$.

By (6), 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 $x \in I_Y \subseteq \Pi_{\mathrm{Y}}$, we have have $h(x) < r - 1/q$ and hence:

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

Again, like in the proof of Claim 3.7, we can increase the entropy gap between $\mathrm{H}(X'|Y')$ and $\mathrm{H}(X|Y)$ to 1. Finally, we apply Proposition A.2 to establish our claim. □

Restatement of Claim 3.9. *For $\Pi \in$ **HV-SZKA**, we can take $I_Y = \emptyset$.*

Proof. 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 $I_Y = \emptyset$, since for every $x \in \Pi_{\mathrm{Y}}$, we have

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

with the last equality following from (6). □

Finally, we prove Proposition 3.11, restated below. Recall that the function $g_x(i, \omega) = (x, i, S(x; \omega)_{2i})$, as stated by (3) in Section 3.3.

Restatement of Proposition 3.11. *Let g_x be as in* (3) *in Section 3.3. For every set $K \subseteq \{0,1\}^*$, if g_x is* not *an instance-dependent distributionally one-way function on K, then for every polynomial p, there exists a nonuniform PPT prover $\widetilde{P}$ such that*

$$\Delta(\langle \widetilde{P}, V\rangle(x), S(x)) \leq \ell(|x|) \cdot \left(\frac{1}{p(|x|)} + 2 \cdot \Delta(\langle P_S, V\rangle(x), S(x))\right) \ ,$$

for infinitely many $x \in K$.

Proof. Let random variable $\mathcal{I}$ denote an independent uniform index $i \leftarrow \{1, 2, \ldots, \ell\}$, and let random variable Ω denote independent uniform coins ω for the simulator S. Recall the definition of instance-dependent distributionally one-way function as stated in Definition 2.3. If g_x is not an instance-dependent distributionally one-way function on K, then for any polynomial q, there exists a nonuniform PPT A such that the random variables $((\mathcal{I}, \Omega), S(x; \Omega)_{2\mathcal{I}})$ and $(A(S(x; \Omega)_{2\mathcal{I}}), S(x; \Omega)_{2\mathcal{I}})$ are $1/q(|x|)$-close for infinitely many $x \in K$. Let $K' \subseteq K$ be the infinite set of instances x for which the previously stated random variables are $1/q(|x|)$-close. Let the polynomial $p(|x|) = q(|x|) \cdot (1/\ell(|x|))$. For this point on, we will drop the mention of x and assume that $x \in K'$.

Since $\mathcal{I}$ is independent from the other random variables, we have that for all $i = 1, 2, \ldots, \ell$, the random variables

$$((i, \Omega), S(\Omega)_{2i}) \text{ and } (A(S(\Omega)_{2i}), S(\Omega)_{2i}) \text{ are } (1/p)\text{-close} \ , \tag{7}$$

since $\ell \cdot (1/q) = 1/p$.

For any interactive machine A and B, let random variable $\langle A, B\rangle[m_j]$ denote the transcript of messages exchanged between A and B conditioned on the first j messages being m_j. In other words, $\langle A, B\rangle[m_j] = \langle A, B\rangle|_{\langle A,B\rangle_j = m_j}$. It follows from definition that

$$\langle A, B\rangle[\langle A, B\rangle_j] \equiv \langle A, B\rangle \ , \tag{8}$$

for any index j.

By (7), and noting that P_S and $\widetilde{P}$ use (i, Ω) and $A(S(\Omega)_{2i})$ to produce their messages in round $2i + 1$, respectively, we have that for every $i = 1, 2, \ldots, \ell$,

$$(\langle P_S, V\rangle[S_{2i}])_{2i+2} \text{ and } (\langle \widetilde{P}, V\rangle[S_{2i}])_{2i+2} \text{ are } (1/p)\text{-close} \ , \tag{9}$$

Using (8) and (9) above, we have that for every $i = 1, 2, \ldots, \ell$,

$$\begin{aligned}
&\Delta(\langle \widetilde{P}, V\rangle_{2i+2}, \langle P_S, V\rangle_{2i+2}) \\
&\quad = \Delta((\langle \widetilde{P}, V\rangle[\langle \widetilde{P}, V\rangle_{2i}])_{2i+2}, (\langle P_S, V\rangle[\langle P_S, V\rangle_{2i}])_{2i+2}) && \text{(by (8))} \\
&\quad \leq \Delta((\langle \widetilde{P}, V\rangle[S_{2i}])_{2i+2}, (\langle P_S, V\rangle[S_{2i}])_{2i+2}) \\
&\qquad + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle_{2i}, S_{2i}) \\
&\quad \leq (1/p) + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle_{2i}, S_{2i}) && \text{(by (9))} \\
&\quad \leq (1/p) + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle, S) \ .
\end{aligned} \tag{10}$$

We now prove the following by induction on $i = 0, 1, 2, \ldots, \ell$:

$$\Delta(\langle \widetilde{P}, V \rangle_{2i}, S_{2i}) \leq i \cdot (1/p + 2 \cdot \Delta(\langle P_S, V \rangle, S)) \tag{11}$$

Note that the case for $i = \ell$ establishes Proposition 3.11. The base case for $i = 0$ is trivial. We prove the inductive step as follows:

$$\begin{aligned}
&\Delta(\langle \widetilde{P}, V \rangle_{2i+2}, S_{2i+2}) \\
&\quad \leq \Delta(\langle \widetilde{P}, V \rangle_{2i+2}, \langle P_S, V \rangle_{2i+2}) + \Delta(\langle P_S, V \rangle_{2i+2}, S_{2i+2}) \\
&\quad \leq \Delta(\langle \widetilde{P}, V \rangle_{2i+2}, \langle P_S, V \rangle_{2i+2}) + \Delta(\langle P_S, V \rangle, S) \\
&\quad \leq (1/p) + \Delta(\langle \widetilde{P}, V \rangle_{2i}, S_{2i}) + 2 \cdot \Delta(\langle P_S, V \rangle, S) && \text{(by (10))} \\
&\quad \leq (i+1) \cdot (1/p + 2 \cdot \Delta(\langle P_S, V \rangle, S)) && \text{(by induction on } i\text{)}
\end{aligned}$$

This completes our proof of Proposition 3.11. □

References

[AH] 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. Extended abstract in *FOCS'87*.

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

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

[BLV] 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*.

[BM] 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.

[BMO] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 482–493, 1990.

[CT] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 2 edition, 2006.

[ESY] 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+] 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.

[For] Lance Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989.

[GK1] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GK2] 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.

[GMR] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Extended abstract in *STOC'85*.

[GMW] 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. Extended abstract in *FOCS'86*.

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

[Gol1] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol2] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[Gol3] 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.

[GS] 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.

[GSV] 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 (STOC)*, pages 399–408, 1998.

[GV] 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.

[HILL] 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. Extended abstracts in *STOC'89* and *STOC'90*.

[HORV] Iftach Haitner, Shien Jin Ong, Omer Reingold, and Salil Vadhan. Instance-dependent commitments for statistical zero knowledge proofs. In preparation, March 2007.

[HR] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. Technical Report 2006/436, Cryptology ePrint Archive, 2006.

[IL] 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 (FOCS)*, pages 230–235, 1989.

[IOS] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *Journal of Cryptology*, 10(1):37–49, 1997.

[MV] Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO)*, pages 282–298, 2003.

[Nao] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. Extended abstract in *CRYPTO'89*.

[NOV] Minh-Huyen 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 (FOCS)*, pages 3–14, 2006.

[NV] Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 287–295, 2006.

[Oka] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000. Extended abstract in *STOC'96*.

[Ost] 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.

[OV] Shien Jin Ong and Salil Vadhan. Zero knowledge and soundness are symmetric. Technical Report TR06-139, Electronic Colloquium on Computational Complexity, 2006. Same version also in Cryptology ePrint Archive, Report 2006/414.

[OW] 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.

[PT] Erez Petrank and Gábor Tardos. On the knowledge complexity of NP. *Combinatorica*, 22(1):83–121, 2002. Extended abstract in *FOCS'96*.

[SV] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003. Extended abstract in *FOCS'97*.

[Vad] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. Extended abstract in *FOCS'04*.