An Unconditional Study of Computational Zero Knowledge*

Salil P. Vadhan†
Division of Engineering & Applied Sciences
Harvard University
Cambridge, MA 02138
salil@eecs.harvard.edu
http://eecs.harvard.edu/~salil/

March 25, 2005

Abstract

We prove a number of general theorems about **ZK**, the class of problems possessing (computational) zero knowledge proofs. Our results are *unconditional*, in contrast to most previous works on **ZK** which rely on the assumption that one-way functions exist.

We establish several new characterizations of **ZK**, and use these characterizations to prove results such as:

1. Honest-verifier **ZK** equals general **ZK**.
2. Public-coin **ZK** equals private-coin **ZK**.
3. **ZK** is closed under union (and more generally, "monotone formula closure").
4. **ZK** with imperfect completeness equals **ZK** with perfect completeness.
5. Any problem in $\mathbf{ZK} \cap \mathbf{NP}$ can be proven in computational zero knowledge by a $\mathbf{BPP}^{\mathbf{NP}}$ prover.
6. **ZK** with black-box simulators equals **ZK** with general, non-black-box simulators.

The above equalities refer to the resulting *class* of problems (and do not necessarily preserve other efficiency measures such as round complexity).

Our approach is to combine the conditional techniques previously used in the study of **ZK** with the unconditional techniques developed in the study of **SZK**, the class of problems possessing statistical zero knowledge proofs. To enable this combination, we prove that every problem in **ZK** can be decomposed into a a problem in **SZK** together with a set of instances from which a one-way function can be constructed.

Keywords: cryptography, computational complexity, zero-knowledge proofs, pseudoentropy, language-dependent commitment schemes, auxiliary-input one-way functions

*An extended abstract of this paper has appeared in *FOCS '04* [Vad2]

†Work done while a Fellow at the Radcliffe Institute for Advanced Study. Also supported by NSF grants CNS-0430336, CCR-0205423, and CCR-0133096, ONR Grant N00014-04-1-0478, and a Sloan Research Fellowship.

Contents

1 Introduction

Since their introduction by Goldwasser, Micali, and Rackoff [GMR], zero-knowledge interactive proofs have become a central tool in cryptographic protocol design. In addition, they have also provided fertile grounds for complexity-theoretic investigations into the interplay between fundamental notions such as proofs, randomness, interaction, and secrecy.

The notion of zero-knowledge proofs raised a number of intriguing basic questions, such as:

- Can we characterize the class **ZK** of problems possessing zero-knowledge proofs?[1] (Here and throughout, we refer to classes of *promise problems* $\Pi = (\Pi_Y, \Pi_N)$ where Π_Y and Π_N are disjoint sets of strings containing the YES instances and NO instances of Π, respectively [ESY]; see Section 2.3.)
- Can we transform proof systems that are zero knowledge for the "honest verifier" (ie the verifier that follows the specified protocol) into ones that are zero knowledge in general (ie for all polynomial-time verifier strategies)? That is, does **HVZK** = **ZK**, where **HVZK** denotes the class of problems possessing honest-verifier zero-knowledge proofs?
- Is it always possible to modify zero-knowledge proofs to have additional useful properties — such as having a small number of rounds, perfect completeness, or public coins? Or do the latter properties restrict the class of problems possessing zero-knowledge proofs?
- What closure properties does **ZK** have? Is it closed under complement? union?

Almost all of these questions were seemingly resolved by a series of exciting works within a few years after zero-knowledge proofs were defined. Specifically, under the assumption that one-way functions exist, it was shown that **ZK** "hits the roof," namely **ZK** = **IP** [GMW, IY, BGG$^+$, Nao, HILL]. Thus, **ZK** is completely characterized and moreover has natural complete problems (namely, any complete problem for **IP** = **PSPACE** [LFKN, Sha]). This also implies that **HVZK** equals **ZK**, since **ZK** $\subseteq$ **HVZK** $\subseteq$ **IP** is immediate from the definitions. In addition, the equality **ZK** = **IP** is proven by a generic transformation from interactive proofs into zero-knowledge proofs, and this transformation preserves many properties such as those mentioned above: the round complexity[2], public coins, and perfect completeness. Since it was known how to transform interactive proofs into ones with public coins [GS] and perfect completeness [FGM$^+$], the same holds follows for zero-knowledge proofs. **ZK** also inherits all the closure properties of **IP** = **PSPACE**, in particular closure under complement and union. However, *all of these results are based on the assumption that one-way functions exist*, and without this assumption, all the questions listed above were open.

In this paper, we answer most of these questions *unconditionally* (i.e., without any unproven complexity assumption). In particular, we:

- Give several characterizations of **ZK** that make no reference to interaction or zero knowledge. (These characterizations are not complete problems, but turn out to have much of the same utility.).

[1] In this paper, we focus on the original notion of *computational* zero-knowledge *proof* systems, as introduced in [GMR]. That is, the zero-knowledge condition is with respect to computationally bounded verifiers (and distinguishers), and the soundness is with respect computationally unbounded prover strategies. In particular, we do not consider *argument* systems, which are only computationally sound.

[2] The round complexity is preserved up to an additive constant for achieving polynomially small soundness error. For negligible error, any superconstant factor suffices (by sequential repetition).

- Prove that **HVZK** = **ZK**.
- Show how to transform any computational zero knowledge proof into one with public coins and perfect completeness.
- Prove that computational zero knowledge is closed under union (and more generally satisfies "monotone formula closure").

This paper is inspired by the work of Ostrovsky and Wigderson [OW], who gave the first hint that it might be possible to prove unconditional results about zero knowledge. They showed that if computational zero knowledge is nontrivial (i.e. **ZK** $\neq$ **BPP**), then "some form of one-way functions" exist. Thus, they made the appealing suggestion that one might prove unconditional results about computational zero knowledge by a case analysis: If **ZK** = **BPP**, then many results about **ZK** hold trivially (because every problem in **BPP** has a trivial zero-knowledge proof where the prover sends nothing and the verifer decides membership on its own using the **BPP** algorithm). On the other hand, if **ZK** $\neq$ **BPP**, then we can try to use their "one-way functions" in the known conditional results about **ZK**. Unfortunately, as they point out, this approach does not work because the form of one-way functions they construct (in case **ZK** $\neq$ **BPP**) are too weak for the conditional constructions mentioned above.[3] A more precise description of the Ostrovsky–Wigderson results and the contrast with ours can be found in Section 7.1.

Our approach is to replace **BPP** by **SZK**, the class of problems possessing *statistical* zero-knowledge proofs (to be described in more detail shortly). In particular, in case **ZK** $\neq$ **SZK**, we are able to construct a form of one-way functions that is much closer to the standard notion than in the Ostrovsky–Wigderson result. However, now the case that **ZK** = **SZK** is not as trivial as before; instead we rely on a large body of previous work giving unconditional results about **SZK** (as described below). To make this approach work, we actually carry out the case analysis on an input-by-input basis. That is, we show that for every problem in **ZK**, we can partition its instances into "**SZK** instances" and "one-way function instances." This characterization is described in more detail below.

1.1 The SZK/OWF Characterization

Statistical Zero Knowledge. The distinction between general (computational) zero knowledge and statistical zero knowledge involves the formulation of the "zero knowledge property," i.e. the requirement that the verifier "learns nothing" from the interaction other than the fact that the assertion being proven is true. The original (and most general) notion discussed above, called *computational zero knowledge*, informally says that a *polynomial-time* verifier learns nothing. *Statistical zero knowledge*, guarantees that even a computationally unbounded verifier learns nothing from the interaction.[4] Naturally, the stronger security guarantee of statistical zero knowledge is preferable,

[3] A similar approach was used in an attempt to prove **HVSZK** = **SZK** [DOY], but subsequently a more direct approach that avoids these difficulties was found [GSV1].

[4] Recall that the zero-knowledge property is formalized by asking that there be probabilistic polynomial-time algorithm S that "simulates" the verifier's view of the interaction (when the assertion being proven is true). In computational zero knowledge, the output distribution of the simulator is only required to be computationally indistinguishable from the verifier's view of the interaction, whereas in statistical zero knowledge, it must be statistically close. We note that there is a similar choice in the soundness condition. We, like [GMR], focus on interactive *proof* systems, where even a *computationally unbounded* prover cannot convince the verifier to accept a false statement, except with negligible probability. In interactive *argument* systems [BCC], this soundness condition is only required for *polynomial-time* provers.

but unfortunately it seems to severely constrain the class of statements that can be proven in zero knowledge. Specifically, it is known that the class **SZK** of problems possessing statistical zero-knowledge proofs is contained in $\mathbf{AM} \cap \mathbf{co\text{-}AM}$ [For, AH], and thus **NP**-complete problems are unlikely to have statistical zero-knowledge proofs. Thus statistical zero-knowledge proofs do not seem to have the wide applicability of computational zero-knowledge proofs (which stems from the existence of computational zero-knowledge proofs for all of **NP** [GMW]).

Nevertheless, the class **SZK** of problems possessing statistical zero-knowledge proofs has turned out to be rich object of study, and in recent years, there have been a number of results substantially improving our understanding it. These results include the identification of natural complete problems for class **SZK** [SV, GV], showing that **SZK** is closed under complement [Oka], honest-verifier **SZK** equals general **SZK** [GSV1], and private-coin **SZK** equals public-coin **SZK** [Oka], and more.[5] In contrast to what was known for computational zero knowledge, all of these results are unconditional. That is, they do not rely on any unproven complexity assumptions (such as the existence of one-way functions).

It was suggested in [Vad1] that the study of **SZK** could provide a useful testbed for understanding zero knowledge, before moving on to more complex models that incorporate computational intractability (such as **ZK**). In this paper, we make extensive use of that methodology, not just proving results about **ZK** by analogy to **SZK**, but actually making direct use of known results about **SZK** (e.g. in establishing and using the characterization below).

The characterization. In this paper, we provide a new characterization of **ZK** in terms of **SZK** and one-way functions:

Definition 1.1 *A promise problem* $\Pi = (\Pi_Y, \Pi_N)$ *satisfies the* SZK/OWF Characterization *if there exists* $I \subseteq \Pi_Y$*, a polynomial-time computable function* $f_x(y) \stackrel{\text{def}}{=} f(x,y)$ *and a polynomial* $p(n)$ *such that the following holds:*

- *Ignoring the inputs in* I*, the problem* Π *has a statistical zero-knowledge proof. Formally, we have* $\Pi' \in \mathbf{SZK}$*, where* $\Pi' = (\Pi_Y \setminus I, \Pi_N)$*.*
- *When* $x \in I$*, the function* f_x *is hard to invert. That is, for every nonuniform polynomial-time algorithm* A*, there exists a negligible function* ϵ *such that for every* $x \in I$*,*

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

Intuitively, this characterization says that for every YES instance x, either one can prove the membership of x in Π in statistical zero knowledge ("x is an **SZK** instance"), or one can use x to construct a one-way function that is given x as an auxiliary input ("x is a OWF instance"). Note that if one-way functions exist (in the standard sense, without auxiliary input), then *all* promise problems satisfy the SZK/OWF Characterization (by setting $I = \Pi_Y$, and $f_x(y) = g(y)$ where g is the one-way function assumed to exist).

On the other hand, **ZK** is contained in **IP**, so the above condition alone *cannot* characterize **ZK** (given the possibility that one-way functions do exist). We prove that if we simply add the condition $\Pi \in \mathbf{IP}$, then we do indeed obtain an exact characterization.

Theorem 1.2 $\Pi \in \mathbf{ZK}$ *if and only if* $\Pi \in \mathbf{IP}$ *and* Π *satisfies the* SZK/OWF Characterization.

[5] See [Vad1] for a unified presentation of all of these results.

One way of thinking of this theorem is as a common generalization of the fact that **SZK** $\subseteq$ **ZK** and the result of [IY, BGG$^+$] that says that the existence of one-way functions implies **ZK** = **IP**. (If one-way functions exist, then every Π satisfies the SZK/OWF Characterization, so Theorem 1.2 becomes $\Pi \in$ **ZK** $\iff$ $\Pi \in$ **IP**.) As noted above, the usefulness of this characterization is that it essentially reduces the unconditional study of **ZK** to its conditional study plus the study of **SZK**.

Theorem 1.2 is in some sense the central theorem of this paper; all of the other results are deduced as consequences of it or its proof.

1.2 Proof Overview

In proving each direction of Theorem 1.2, we actually prove stronger statements than required. In the forward ("only if") direction, we actually show that every problem in **HVZK**, not just **ZK**, satisfies the SZK/OWF Characterization. In the reverse ("if") direction, we show that every problem in **IP** satisfying the SZK/OWF Characterization is not only in **ZK**, but has a computational zero-knowledge proof with many nice properties, such as public coins, perfect completeness, universal black-box simulation, etc. Combining the two directions, we deduce that **HVZK** = **ZK**, and that every problem in **ZK** has a computational zero-knowledge proof with the aforementioned properties.

From HVZK to the SZK/OWF Characterization**.** In proving this direction, we first establish intermediate characterizations of **HVZK** that are computational analogues of the complete problems for **SZK** [SV, GV]. The reductions from **HVZK** to these intermediate characterizations are naturally adaptations of the reductions from **HVSZK** = **SZK** to the **SZK**-complete problems (which in turn are based on the simulator analyses of [For, AH, PT]). The reduction from the intermediate characterizations to the SZK/OWF Characterization combines on a reduction to one of the complete problems for **SZK** (for the "**SZK** instances") with the techniques of Håstad, Impagliazzo, Levin, and Luby [HILL] (for the "OWF instances").

From the SZK/OWF Characterization **to ZK.** Here the goal is to construct a computational zero-knowledge proof system for every problem $\Pi \in$ **IP** that satisfies the SZK/OWF Characterization. A first approach is for the prover to use the **SZK** proof system when the input is in $\Pi_Y \setminus I$, and to use the proof system obtained by the generic, one-way-function-based compiler from **IP** to **ZK** [IY, BGG$^+$] when the input is in I. The difficulty with this is that the set I may not be efficiently recognizable, so this approach leaks information to the verifier (namely whether or not the input is in I). Because of this difficulty, we take a more indirect approach. Instead of trying to construct separate zero-knowledge proofs for the **SZK** instances and the OWF instances and then combine them, we instead construct a type of bit-commitment scheme in each of the two cases. The advantage is that the commitment schemes are easy to combine. We then use the combined commitment scheme in the generic compiler from **IP** to **ZK** [IY, BGG$^+$].

The type of commitment schemes we construct are *problem-dependent commitment schemes* [BMO, IOS, MV]. For a promise problem Π, in a Π-*dependent commitment scheme*, both the sender and receiver get an common auxiliary input x, which is an instance of Π. If x is a YES instance of Π, then the protocol is hiding, and if x is a NO instance, then the protocol is binding. Thus, they are a relaxation of commitment schemes, because the hiding and binding properties are not required to hold at the same time. Nevertheless, this relaxation is still useful in constructing zero-knowledge

proofs. The reason is that zero-knowledge proofs based on commitments (e.g. [GMW, IY, BGG$^+$]) typically only use the hiding property in proving zero knowledge (which is only required when x is a YES instance) and the binding property in proving soundness (which is only required when x is a NO instance).

We show that every problem satisfying the SZK/OWF CHARACTERIZATION has a problem-dependent commitment scheme. This is done by combining two problem-dependent commitment schemes, one that is hiding on the OWF instances (using [Nao, HILL]) and the other that is hiding on the **SZK** (YES) instances. (Both are binding on NO instances.) The construction of the problem-dependent commitment scheme for **SZK** is based on a combination of techniques from [Oka, SV, GV], and is the technically most involved portion of our proof.

Putting together all of our results, we deduce that the SZK/OWF CHARACTERIZATION, the computational analogues of the **SZK**-complete problems, and the problem-dependent commitments, all provide equivalent characterizations of $\mathbf{ZK} = \mathbf{HVZK}$. These characterizations may be of independent interest.

1.3 Organization

We begin in Section 2 with definitions, notations, and basic results we will use throughout the paper, in particular covering probability and information theory, promise problems, and zero-knowledge proofs. Section 3 contains the proof of the forward direction of Theorem 1.2, including establishing the computational analogues of the **SZK**-complete problems. Section 4 contains the proof of the reverse direction of Theorem 1.2, except for the construction of problem-dependent commitments for all of **SZK**, which is deferred to Section 5. Section 6 ties together the results of Sections 3–5, in particular establishing Theorem 1.2. Section 7 contains several applications and extensions of our results, including monotone closure properties of **ZK**, new proofs of the Ostrovsky–Wigderson Theorem, and an equivalence between strict and expected polynomial-time simulators. In Section 8, we conclude with some open problems and directions for further work.

2 Preliminaries

2.1 Basic Notation

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 $\{0,1\}^n$. The *support* of a random variable X is $\mathrm{Supp}(X) = \{x : \Pr[X = x] > 0\}$. A random variable is *flat* if it is uniform over its support. If X and Y are random variables, then $X \otimes Y$ denotes the random variable obtained by taking independent random samples $x \leftarrow X$ and $y \leftarrow Y$ and outputting (x, y). We write $\otimes^k X$ to denote the random variable consisting of k independent copies of X. For an event E, $X|_E$ denotes the random variable X conditioned on E.

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

negligible function $\mu(n)$ such that for every n, $f(n) < \mu(n)$). Likewise, poly(n) denotes an arbitrary polynomial.

For a probabilistic algorithm A, we write $A(x;r)$ to denote the output of A on input x and coin tosses r. $A(x)$ is a random variable denoting the output of A for uniformly selected coin tosses. *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 series 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 families of polynomial-sized Boolean circuits.

A circuit $C : \{0,1\}^m \to \{0,1\}^n$ defines a probability distribution on $\{0,1\}^n$ by evaluating C on a uniformly chosen input in $\{0,1\}^m$. That is, we view C as specifying a sampling algorithm for the distribution, with its input gates being the coin tosses; thus we will often refer to distributions specified by circuits as *samplable distributions*. These will play a central role in the paper.

2.2 Statistical Measures

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

$$\begin{aligned}\Delta(X,Y) &= \max_{S \subset \mathcal{U}} |\Pr[X \in S] - \Pr[Y \in S]| \\ &= \frac{1}{2}\sum_{x \in \mathcal{U}} |\Pr[X = x] - \Pr[Y = x]| \\ &= 1 - \sum_{x \in \mathcal{U}} \min\{\Pr[X = x], \Pr[Y = y]\}\end{aligned}$$

We say that X and Y are ε-*close* if $\Delta(X,Y) \leq \varepsilon$. For basic facts about this metric, see [SV, Sec 2.3].

Entropy. The *entropy* of a random variable X is $\mathrm{H}(X) = \mathrm{E}_{x \leftarrow X}[\log(1/\Pr[X = x])]$, where here and throughout the paper all logarithms are base 2. Intuitively, $\mathrm{H}(X)$ measures the amount of randomness in X *on average* (in bits). The *min-entropy* of X is $\mathrm{H}_\infty(X) = \min_x[\log(1/\Pr[X = x])]$; this is a "worst-case" measure of randomness. In general $\mathrm{H}_\infty(X) \leq \mathrm{H}(X)$, but if X is flat, then $\mathrm{H}(X) = \mathrm{H}_\infty(X) = \log|\mathrm{Supp}(X)|$. For $p \in [0,1]$, we define the *binary entropy function* $\mathrm{H}_2(p)$ to be the entropy of a binary random variable that is 1 with probability p and 0 with probability $1-p$, i.e. $\mathrm{H}_2(p) = p \cdot \log(1/p) + (1-p) \cdot \log(1/(1-p))$. For jointly distributed random variables X and Y, we define the *conditional entropy of X given Y* to be

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

A useful fact is that if two random variables are statistically close, then their entropies must also be close:

Lemma 2.1 (cf., [GV, Fact B.1]) *For any two random variables, X and Y, ranging over a universe $\mathcal{U}$ it holds that*

$$|\mathrm{H}(X) - \mathrm{H}(Y)| \;\leq\; \log(|\mathcal{U}| - 1) \cdot \delta \;+\; \mathrm{H}_2(\delta)$$

where $\delta \stackrel{\text{def}}{=} \Delta(X,Y)$.

For more background on entropy, see [CT].

Direct Products. We will often refer to the behavior of the above measures under direct products, i.e. when we take k independent copies of a random variable. For statistical difference, we have the following bounds:

Lemma 2.2 (cf., [SV, Lemma 3.4]) *For random variables X and Y, if $\delta = \Delta(X,Y)$, then for every $k \in \mathbb{N}$, we have*

$$k\delta \geq \Delta(\otimes^k X, \otimes^k Y) \geq 1 - 2\exp(-k\delta^2/2).$$

For entropy, it holds that for every X, Y, $\mathrm{H}(X \otimes Y) = \mathrm{H}(X) + \mathrm{H}(Y)$ and thus $\mathrm{H}(\otimes^k X) = k \cdot \mathrm{H}(X)$. Similarly, for conditional entropy, if we write $\otimes^k(X,Y) = ((X_1,Y_1),\ldots,(X_k,Y_k))$, then $\mathrm{H}((X_1,\ldots,X_k)|(Y_1,\ldots,Y_k)) = k \cdot \mathrm{H}(X|Y)$.

Another well-known and useful feature of taking direct products is that it makes "flattens" random variables, so that probability masses become concentrated around $2^{-\mathrm{H}(X)}$. (This is known as the Asymptotic Equipartition Property in information theory; see [CT].) Our formalization of it follows [GV], with an extension to conditional distributions.

Definition 2.3 (heavy, light and typical elements) *Let X be a random variable taking values in a universe $\mathcal{U}$, x an element of $\mathcal{U}$, and Δ a positive real number. We say that x is* Δ-heavy *(resp.,* Δ-light*) if* $\Pr[X = x] \geq 2^{\Delta} \cdot 2^{-\mathrm{H}(X)}$ (resp., $\Pr[X = x] \leq 2^{-\Delta} \cdot 2^{-\mathrm{H}(X)}$). *Otherwise, we say that x is* Δ-typical.

If Y is a random variable jointly distributed with X, and $y \in \mathrm{Supp}(Y)$, we say that x is Δ-heavy given y *(resp.,* Δ-light given y *if* $\Pr[X = x|Y = y] \geq 2^{\Delta} \cdot 2^{-\mathrm{H}(X|Y)}$ (resp., $\Pr[X = x|Y = y] \leq 2^{-\Delta} \cdot 2^{-\mathrm{H}(X|Y)}$). *Otherwise, we say that x is* Δ-typical given y.

A natural relaxed definition of flatness follows. The definition links the amount of slackness allowed in "typical" elements with the probability mass assigned to non-typical elements.

Definition 2.4 (nearly flat distributions) [6] *A distribution X is called* Δ-flat *if for every $t \geq 1$ the probability that an element chosen from X is $t \cdot \Delta$-typical is at least $1 - 2^{-t^2}$.*

If Y is jointly distributed with X, then we say that X is Δ-flat given Y *if for every $t \geq 1$, when $(x,y) \leftarrow (X,Y)$, the probability that x is $t \cdot \Delta$-typical given y is at least $1 - 2^{-t^2}$.*

A consequence of this definition is that if X is Δ-flat, then for every $t \geq 1$, X is 2^{-t^2}-close to a random variable X' with *min-entropy* at least $\mathrm{H}(X) - t\Delta$.

Lemma 2.5 (Flattening Lemma) *Let X be a distribution, k a positive integer, and $\otimes^k X$ denote the distribution composed of k independent copies of X. Suppose that for all x in the support of X it holds that $\Pr[X = x] \geq 2^{-m}$. Then $\otimes^k X$ is $\sqrt{k} \cdot m$-flat.*

Suppose Y is jointly distributed with X, and for all (x,y) in the support of (X,Y) it holds that $\Pr[X = x|Y = y] \geq 2^{-m}$. Then, defining $((X_1,Y_1),\ldots,(X_k,Y_k)) = \otimes^k(X,Y)$, the random variable $(X_1,\ldots,X_k)$ is $\sqrt{k} \cdot m$-flat given $(Y_1,\ldots,Y_k)$.

The key point is that deviation from flatness grows sublinearly with k, while the entropy grows linearly with k. We prove the Flattening Lemma in Appendix A for completeness.

[6] The definition in [GV] allows any $t > 0$, but only requires that the probability of being $t\Delta$-typical to be $1 - 2^{-t^2+1}$. We find it more convenient to restrict to $t \geq 1$, a setting that is satisfied in all our applications.

Hashing. The topic of *randomness extraction* is concerned with efficiently extracting as many almost-uniform random bits as possible from non-uniformly distributed random variables. The entropy of a random variable does not provide a good measure of how many almost-uniform bits can be extracted, but its min-entropy does, provided we are willing for the extraction procedure itself to be probabilistic. In particular, the Leftover Hash Lemma of [BBR, HILL] shows that universal (or pairwise independent) hash functions can be used for this purpose. (We could use any randomness extractor in the sense of [NZ] with similar parameters, but we restrict to universal hashing for simplicity.)

Lemma 2.6 (Leftover Hash Lemma) *Let H be a randomly selected from a family of universal hash functions mapping $\{0,1\}^n$ to $\{0,1\}^m$. Then, for every $\varepsilon > 0$ and every distribution X on $\{0,1\}^n$ of min-entropy at least $m + 2\log(1/\varepsilon)$, the random variable $(H, H(X))$ is ε-close to (H, U_m).*

Recall that for every n, m, there is an explicit family of universal hash functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, where a random hash function in the family can be described by $O(n+m)$ random bits and can be evaluated in time $\mathrm{poly}(n,m)$.

2.3 Promise Problems

Roughly speaking, a *promise problem* [ESY] is simply 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. (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.

A promise problem Π *reduces* to promise problem Γ if there is a polynomial-time computable function f such that

$$x \in \Pi_Y \Rightarrow f(x) \in \Gamma_Y$$
$$x \in \Pi_N \Rightarrow f(x) \in \Gamma_N.$$

That is we work with polynomial-time mapping reductions (i.e. Karp reductions), unless otherwise specified. If $\mathbf{C}$ is a class of promise problems, then Π is *complete for* $\mathbf{C}$ (or $\mathbf{C}$-*complete*) if $\Pi \in \mathbf{C}$ and every promise problem in $\mathbf{C}$ reduces to Π. Sometimes we will restrict to reductions f that are *non-shrinking*, in the sense that there is a constant $\delta > 0$ such that $|f(x)| \geq |x|^\delta$ for all strings x.

We refer the reader to the recent survey of Goldreich [Gol4] for more on the utility and subtleties of promise problems.

2.4 Auxiliary-Input Cryptographic Primitives

It will be very useful for us to work with cryptographic primitives that are parameterized by an additional "auxiliary" input x, and where the security condition will hold only if x is in some particular set I. Indeed, recall that the SZK/OWF Characterization refers to such a variant of the notion of one-way functions, as captured by the following definitions.

Definition 2.7 *An* auxiliary-input function ensemble *is a collection of functions* $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}_{x\in\{0,1\}^*}$*, where* p *and* q *are polynomials. We call* $\mathcal{F}$ polynomial-time computable *(or just* poly-time*), if there is a (deterministic) polynomial-time algorithm* F *such that for every* $x \in \{0,1\}^*$ *and* $y \in \{0,1\}^{p(|x|)}$*, we have* $F(x,y) = f_x(y)$*.*

Definition 2.8 *An* auxiliary-input one-way function on I *is a poly-time auxiliary-input function ensemble* $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ *such that for every nonuniform PPT* A*, there exists a negligible function* μ *such that for all* $x \in I$*,*

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

(We note that since A is non-uniform, it is not essential that we give it the input x, but it is more natural to think of A as having the input x and it helps ensure that A has sufficient running time even if f_x shrinks its input.) The standard definition of one-way function is obtained by considering $I = \{1^n : n \geq 0\}$ and $p(n) = n$. The above is a stronger notion of auxiliary-input one-way function than the one considered by Ostrovsky and Wigderson [OW]. In their formulation (denoted $\exists 1WF$), the set I is not fixed for all A, but rather can depend on A. That is, they require that for every PPT A, there exists an infinite set I_A such that A has small probability of inverting f_x for all $x \in I_A$. (See Theorem 7.1 for a precise formulation of this notion and the result of [OW].)

Given this definition, we can restate the SZK/OWF Characterization as follows.

Definition 2.9 *A promise problem* $\Pi = (\Pi_Y, \Pi_N)$ *satisfies the* SZK/OWF Characterization *if there is* $I \subseteq \Pi_Y$ *such that:*

- *The promise problem* $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ *is in* **SZK**.
- *There exists an auxiliary-input one-way function on* I*.*

Similarly, the notion of computational indistinguishability has an auxiliary-input analogue (which is widely used in the definition of zero knowledge; see Section 2.5).

Definition 2.10 *An* auxiliary-input probability ensemble *is a collection of random variables* $\{X_x\}_{x\in\{0,1\}^*}$*, where* X_x *takes values in* $\{0,1\}^{p(|x|)}$ *for some polynomial* p*. We call such an ensemble* samplable *if there is a probabilistic polynomial-time algorithm* M *such that for every* x*,* $S(x)$ *is distributed according to* X_x*.*

Definition 2.11 *Two auxiliary-input probability ensembles $\{X_x\}$ and $\{Y_x\}$ are* computationally indistinguishable *on $I \subseteq \{0,1\}^*$ if for every nonuniform PPT D, there exists a negligible function μ such that for all $x \in I$,*

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

Similarly, we say that that X_x and Y_x are statistically indistinguishable *on $I \subseteq \{0,1\}^*$ if the above is required for all functions D (instead of only nonuniform PPT). Equivalently, X_x and Y_x are $\mu(|x|)$-close for some negligible function μ and all $x \in I$. If X_x are Y_x are identically distributed for all $x \in I$, we say that they are* perfectly indistinguishable.

Often, we will informally say "X_x and Y_x are computationally indistinguishable when $x \in I$" to mean the ensembles $\{X_x\}$ and $\{Y_x\}$ are computationally indistinguishable on I. It is well-known that indistinguishability is preserved when we take polynomially many direct products. That is:

Lemma 2.12 (cf., [Gol3, Ch.3, Ex.9]) *If $\{X_x\}$ and $\{Y_x\}$ are computationally indistinguishable on I, then for every polynomial p, $\{\otimes^{p(|x|)} X_x\}$ and $\{\otimes^{p(|x|)} Y_x\}$ are computationally indistinguishable on I.*

Now we can naturally define auxiliary-input pseudorandom generators.

Definition 2.13 *An* auxiliary-input pseudorandom generator on I *is a poly-time auxiliary-input function ensemble $\mathcal{G} = \{G_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ such that $q(n) > p(n)$, and the probability ensembles $\{G_x(U_{p(|x|)})\}_x$ and $\{U_{q(|x|)}\}_x$ are computationally indistinguishable on I.*

Almost all reductions between cryptographic primitives immediately extend to their auxiliary-input analogues (using the same proof). For example:

Theorem 2.14 ([HILL]) *For every set $I \subseteq \{0,1\}^*$, there exists a pseudorandom generator on I if and only if there exists a one-way function on I.*

2.5 Zero-knowledge Proofs

We follow the standard definitions of zero-knowledge interactive proofs, as in [Gol3], noting the following points:

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^*.

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

- *(Efficiency) (P, V) is polynomially bounded, and V is computable in probabilistic polynomial time.*
- *(Completeness) If $x \in \Pi_Y$, then V accepts in $(P, V)(x)$ with probability at least $1 - c(|x|)$,*
- *(Soundness) If $x \in \Pi_N$, 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 $c = 0$.* **IP** *denotes the class of promise problems possessing interactive proof systems.*

We write $\langle A(a), B(b)\rangle(x)$ to denote B's view of the interaction, i.e. a transcript $(\gamma_1, \gamma_2, \ldots, \gamma_t; r)$, where the γ_i's are all the messages exchanged and r is B's coin tosses.

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 verifier that follows the specified protocol.

Definition 2.16 (honest-verifier zero knowledge) *An interactive proof system (P, V) for a promise problem Π is* (perfect/statistical/computational) honest-verifier zero knowledge *if there exists a probabilistic polynomial-time* simulator *S such that the ensembles $\{\langle P, V\rangle(x)\}_{x\in\Pi_Y}$ and $\{S(x)\}_{x\in\Pi_Y}$ are (perfectly/statistically/computationally) indistinguishable. We will often drop the word "computational" in reference to computational zero knowledge.*

HVPZK, **HVSZK**, *and* **HVZK** *denote the classes of promise problems have honest-verifier perfect, statistical, and computational zero-knowledge proofs, 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 2.17 (auxiliary-input zero knowledge[7]) *An interactive proof system (P, V) for a promise problem Π is (perfect/statistical/computational)* (auxiliary-input) zero knowledge *if for every PPT V^* and polynomial p, there exists a PPT S such that the ensembles*

$$\{\langle P, V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^{p(|x|)}} \qquad \textit{and} \qquad \{S(x, z)\}_{x\in L, z\in\{0,1\}^{p(|x|)}} \tag{1}$$

are (perfectly/statistically/computationally) indistinguishable.

PZK, **SZK**, *and* **ZK** *are the classes of promise problems possessing perfect, statistical, and computational (auxiliary-input) zero-knowledge proofs, respectively.*

The auxiliary input z in the above definition models a priori information that the verifier may possess before the interaction begins. Thus auxiliary-input zero knowledge is usually necessary when zero-knowledge proofs are to be used as a subprotocol in a larger protocol, or even when composing zero-knowledge proofs with themselves. Indeed, it is known that auxiliary-input zero knowledge is closed under sequential composition [GO], but plain zero knowledge (i.e. without auxiliary inputs) is not [GK3]. For this reason, auxiliary-input zero knowledge is the definition typically used in the literature.

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 auxililary input), and instead only requires black-box access to V^*. This type of simulation is formalized as follows.

Definition 2.18 (black-box zero knowledge) *We say that (P, V) is (perfect/statistical/computational)* black-box zero knowledge *if there exists an oracle PPT S such that for every nonuniform PPT V^*, the ensembles*

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

are (perfectly/statistically/computationally) indistinguishable.

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 as advice). The recent work of Barak [Bar] demonstrated that non-black-box zero-knowledge proofs can achieve properties (such as simultaneously being public coin and having a constant number of rounds) that were known to be impossible for black-box zero knowledge [GK3]. Nevertheless, our results will show that, when ignoring efficiency considerations, black-box zero knowledge is as rich as standard, auxiliary-input zero knowledge; that is, every problem in **ZK** has a black-box zero-knowledge proof system.

Remarks on the definitions. Our definitions mostly follow the now-standard definitions of zero-knowledge proofs as presented in [Gol3], but we highlight the following points:

1. (Promise problems) As has been done numerous times before (e.g. [GK4, SV]), we extended 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. $\mathbf{ZK}, \mathbf{SZK}, \mathbf{HVZK}, \mathbf{HVSZK}, \mathbf{BPP}$) 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 **ZK** automatically extend to the class of languages, simply because languages are a special case of promise problems.

2. (Nonuniform formulation) As has become standard, we have adopted a nonuniform formulation of zero knowledge, where the computational indistinguishability to hold even with respect to nonuniform distinguishers and is universally quantified over all YES instances. Uniform treatments of zero knowledge are possible (see [Gol2] 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.

3. (Strict polynomial-time simulators) For simplicity, we initially present our results restrict our attention to zero knowledge with respect to simulators that run in *strict* polynomial time. The original definition of zero knowledge [GMR] allows for simulators that run in *expected* polynomial time. In Section 7.3, we extend our techniques to zero knowledge with respect to expected polynomial-time simulators (in fact an even weaker notion), and ultimately prove that the class of problems having zero-knowledge proofs with expected polynomial-time simulators and the class of problems having zero-knowledge proofs with strict polynomial-time simulators are equal.

4. (Proof systems versus arguments) We restrict our attention to the original notion of interactive *proof* systems [GMR, BM], where the soundness condition holds even for computationally unbounded prover strategies. A direction for future work is to obtain similar results for the more general notion of interactive *argument* systems [BCC], where the soundness condition is only required for polynomial-time prover strategies.

5. (Security parameterization) In the definition of computational indistinguishability (Definition 2.11) and consequently in the formulation of the zero-knowledge conditions above, computational indistinguishability is measured in terms of the input length $|x|$. That is, only "long" statements can be proven with "high" security. An alternative and perhaps more natural formulation of zero knowledge (see [Vad1, Section 2.3]) measures indistinguishability in terms of a separate security parameter k, given to the prover, verifier, and simulator, and such that the protocol is allowed running time $\mathrm{poly}(|x|, k)$. We stick with the formulation in terms of the input length $|x|$ because it is the original and more commonly used definition. Moreover, all of our results also hold for the security-parameterized definition, via the observation that a security-parameterized zero-knowledge proof for a promise problem Π is equivalent to a (standard, non-security-parameterized) zero-knowledge proof for the promise problem Π' defined by $\Pi'_Y = \{(x, 1^k) : x \in \Pi_Y, k \in \mathbb{N}\}$ and $\Pi'_N = \{(x, 1^k) : x \in \Pi_N, k \in \mathbb{N}\}$. Note that this does not imply (nor do we claim) that every problem in **ZK** has a security-parameterized zero-knowledge proof. (For **SZK**, however, it was shown in [SV] that every problem in **SZK** has a security-parameterized statistical zero-knowledge proof.)

6. (Closure under reductions) All of the zero-knowledge classes defined above, in particular **HVZK** and **ZK**, are easily seen to be closed under *non-shrinking* reductions f (i.e. ones where $|f(x)| \geq |x|^{\Omega(1)}$): if f reduces Π to $\Gamma \in \mathbf{ZK}$, we can obtain a zero-knowledge proof for Π by having the prover and verifier on input x, execute the zero-knowledge proof for Γ on $f(x)$. The non-shrinking condition is needed because the security of the zero-knowledge proof for Γ is measured as a function of $|f(x)|$, and we need to relate it to security in terms of $|x|$. The non-shrinking condition is unnecessary if one works with a security-parameterized definition of zero-knowledge proofs (cf., [Vad1, Prop. 2.4.1]).

3 From HVZK to the SZK/OWF Characterization

In this section, we prove that every problem in **HVZK** satisfies the SZK/OWF Characterization.

A first attempt. To show that every $\Pi \in \mathbf{HVZK}$ satisfies the SZK/OWF Characterization, it is tempting to take the following approach. Consider the (honest-verifier) simulator for Π's computational zero-knowledge proof system. Let I be the set of inputs $x \in \Pi_Y$ for which the simulator's output is statistically far from the verifier's view. When we ignore the inputs in I, we have an (honest-verifier) statistical zero-knowledge proof system. On I, the output of the simulator and the verifier's view are statistically far apart but computationally indistinguishable. By Goldreich [Gol1], from any two samplable distributions that are statistically far apart but computationally indistinguishable, we can construct a one-way function.

This approach has two difficulties:

- What threshold of statistical difference should we use to partition the inputs in Π_Y? The result of Goldreich requires statistical difference at least $1/p(n)$ for any fixed polynomial $p(n)$, but the definition statistical zero knowledge requires negligible statistical difference $1/n^{\omega(1)}$.
- The result of Goldreich [Gol1] requires that both distributions be (polynomial-time) *samplable*, but the verifier's view of the interaction with the prover will typically not be samplable. Moreover, if we require only one of the two distributions in Goldreich's hypothesis to be samplable, then it is unlikely to imply one-way functions. Indeed, it has been proven unconditionally that the uniform distribution (which is trivially samplable) is computationally indistinguishable from some (non-samplable) distributions that are statistically very far from uniform (indeed have very low entropy) [GK2].

The first difficulty can be overcome using known results about **SZK**. Specifically, in [GV] it is shown that if a problem Π has an interactive proof system that can be simulated within statistical difference within $1/p(n)$ for a sufficiently large (but fixed) polynomial p (e.g. the cube of the communication complexity), then $\Pi \in \mathbf{SZK}$.

For the second difficulty, we use the fact that a samplable distribution that is computationally indistinguishable from a non-samplable distribution of noticeably *higher entropy* does imply one-way functions [HILL]. This leads us to look for "high-entropy" distributions in the real prover-verifier interaction. We find such distributions using the techniques of [AH, PT, GV]. This approach leads us to establish two other characterizations of **ZK** en route to the SZK/OWF Characterization. These characterizations are computational analogues of the complete problems for **SZK**, and may be of independent interest.

3.1 Analogues of the SZK-Complete Problems

We establish two characterizations of **ZK** that are related to the the complete problems for **SZK**, so we begin by recalling those.

The Complete Problems for SZK. The first problem is Statistical Difference, the promise problem $\mathrm{SD} = (\mathrm{SD}_Y, \mathrm{SD}_N)$ defined by

$$\begin{aligned} \mathrm{SD}_Y &= \{(X,Y) : \Delta(X,Y) \leq 1/3\} \\ \mathrm{SD}_N &= \{(X,Y) : \Delta(X,Y) \geq 2/3\}, \end{aligned}$$

where X and Y are probability distributions specified by circuits that sample from them, and $\Delta(X,Y)$ denotes statistical difference. (See Sections 2.1 and 2.2.)

The second problem is Entropy Difference, the promise problem $\mathrm{ED} = (\mathrm{ED}_Y, \mathrm{ED}_N)$ defined by

$$\begin{aligned} \mathrm{ED}_Y &= \{(X,Y) : \mathrm{H}(X) \geq \mathrm{H}(Y) + 1\} \\ \mathrm{ED}_N &= \{(X,Y) : \mathrm{H}(X) \leq \mathrm{H}(Y) - 1\}, \end{aligned}$$

where $\mathrm{H}(\cdot)$ denotes the entropy function (see Section 2.2).

The Completeness Theorems of [SV, GV] can be stated as follows.

Theorem 3.1 ([SV, GV]) Statistical Difference *and* Entropy Difference *are complete for* **SZK**. *That is, they are both in* **SZK** *and for every problem* $\Pi \in$ **SZK**, *there is a polynomial-time computable function mapping strings* x *to pairs of samplable distributions* (X, Y) *such that*

- *If* $x \in \Pi_Y$, *then* $\Delta(X,Y) \leq 1/3$,
- *If* $x \in \Pi_N$, *then* $\Delta(X,Y) \geq 2/3$,

Similarly for Entropy Difference.

Note that the result that **SZK** is closed under complement [Oka] follows from the fact Entropy Difference trivially reduces to its complement (via the reduction $(X,Y) \mapsto (Y,X)$).

Analogous Characterizations of ZK. We present analogous characterizations of **ZK**, albeit not in terms of complete problems.

Definition 3.2 *A promise problem* Π *satisfies the* Indistinguishability Characterization *if there is a polynomial-time computable function mapping strings* x *to pairs of samplable distributions* (X, Y) *such that*

- *If* $x \in \Pi_Y$, *then* X *and* Y *are computationally indistinguishable (in the sense of Definition 2.11),*
- *If* $x \in \Pi_N$, *then* $\Delta(X,Y) \geq 2/3$.

We note that the constant $2/3$ in Item 3.2 is arbitrary. By taking direct products and applying Lemmas 2.2 and 2.12, we can boost a threshold as low as $1/\mathrm{poly}(n)$ to as high as $1 - 2^{-\mathrm{poly}(n)}$, while preserving the computational indistinguishability in Item 3.2.

Like the SZK/OWF Characterization, if one-way functions exist, then every promise problem satisfies the Indistinguishability Characterization: on an input x of length n, we can define $X = G(U_n)$ and $Y = U_{2n}$, where G is a length-doubling pseudorandom generator, and then X and Y simultaneously are computationally indistinguishable and have large statistical difference. Thus, as before, to obtain a characterization of **ZK**, we need to add the condition $\Pi \in$ **IP**.

This example also illustrates why Π satisfying the Indistinguishability Characterization cannot be cast as a reduction from Π to some promise problem — the conditions for yes instances and no instances may hold at the same time. Nevertheless, the Indistinguishability Characterization turns out to have much of the same utility as a complete problem (like Statistical Difference).

Our results will imply the following theorem.

Theorem 3.3 $\Pi \in$ **ZK** *if and only if* $\Pi \in$ **IP** *and* Π *satisfies the* Indistinguishability Characterization.

In [SV], it was already proven that every problem that has a *public-coin* computational zero-knowledge proof satisfies the Indistinguishability Characterization. Thus, what is new here is showing that the characterization *holds even for private-coin proofs*, and *establishing a converse* (for $\Pi \in$ **IP**).

A characterization somewhat analogous to Entropy Difference follows.

Definition 3.4 *A promise problem* Π *satisfies the* Conditional Pseudoentropy Characterization *if there is a polynomial-time computable function mapping strings* x *to a* samplable joint distribution (X, Y) *(i.e. two circuits that use the same coin tosses) and a parameter* r *such that*

- *If* $x \in \Pi_Y$*, then there exists a (not necessarily samplable) joint distribution* (X', Y') *such that* (X', Y') *is computationally indistinguishable from* (X, Y) *and* $\mathrm{H}(X'|Y') \geq r$*, and*
- *If* $x \in \Pi_N$*, then* $\mathrm{H}(X|Y) \leq r - 1$*,*

where $\mathrm{H}(\cdot|\cdot)$ *denotes conditional entropy. (See Section 2.2.)*

As before, this definition is satisfied by all promise problems if one-way functions exist. A crucial point is that we use the conditional entropy $\mathrm{H}(X|Y)$ instead of the difference in entropies $\mathrm{H}(X) - \mathrm{H}(Y)$ (as in the definition of Entropy Difference). Indeed, in [Vad1] we pointed out that the analogous condition using difference in entropies is satisfied by *all* promise problems (regardless of whether or not one-way functions exist) and thus is useless.[8] (At the time, we saw this as an obstacle to finding **ZK** analogues of the complete problems for **SZK**.) Our use of conditional entropy was inspired in part by its role in the the conjectures of [BLV, Sec. 9].

Theorem 3.5 $\Pi \in$ **ZK** *if and only if* $\Pi \in$ **IP** *and* Π *satisfies the* Conditional Pseudoentropy Characterization.

Note that, in contrast to the **SZK**-completeness of Entropy Difference, this theorem does not seem to imply that **ZK** is closed under complement. The reason is that the Conditional Pseudoentropy Characterization is not symmetric with respect to yes and no instances.

In the remainder of this section, we will show that every promise problem in **HVZK** satisfies the Conditional Pseudoentropy Characterization, the Indistinguishability Characterization, and the SZK/OWF Characterization. This establishes the forward ("only if") directions of Theorems 1.2, 3.3, and 3.5. The reverse directions, showing that problems in **IP** satisfying the characterizations are in **ZK**, will be done in Section 4.

[8] The reason comes from the fact that we do not require X' and Y' above to be samplable. It is known (via the Probabilistic Method) that there exist (non-samplable) low-entropy distributions that are indistinguishable from the uniform distribution [GK2]. Thus, if the above characterization referred to $\mathrm{H}(X) - \mathrm{H}(Y)$, then it would hold for all promise problems, by setting $X' = X$ and Y' to be some low-entropy distribution indistinguishable from $Y = U_n$.

3.2 The Conditional Pseudoentropy Characterization

Lemma 3.6 *If a promise problem* Π *is in* **HVZK**, *then* Π *satisfies the* Conditional Pseudoentropy Characterization.

Proof: The proof is an adaptation of the reduction from **HVZK** to Entropy Difference in [GV]. Let (P, V) be an honest-verifier computational zero-knowledge proof for Π, with simulator S. We modify the proof system to satisfy the following (standard) additional properties:

- The completeness error $c(|x|)$ and soundess 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. Having the verifier send its coin tosses at the end does not affect soundness because it is after the prover's last message, and does not affect honest-verifier zero knowledge because the simulator is anyhow required to simulate the verifier's coin tosses.
- On every input x, the simulator always outputs *accepting transcripts*, where we call a sequence γ of 2ℓ messages an accepting transcript on x if all of the verifier's messages are consistent with its coin tosses (as specified in the last message), and the verifier would accept in such an interaction. To achieve this, we first modify the proof system so that the verifier always accepts if its coin tosses are $0^{r(|x|)}$; this increases the soundness error only negligibly. Then we modify the simulator so that any time it is about to output a non-accepting transcript, it instead outputs the accepting transcript where all of the prover messages are the empty string and the verifier's coin tosses are $0^{r(|x|)}$. This has a negligible effect the quality of the simulation because when $x \in Pi_Y$, the original simulator could only output non-accepting transcripts with negligible probability (otherwise its output could easily be distinguished from the real interaction, which has non-accepting transcripts with probability at most $c(|x|) = \mathrm{neg}(|x|)$).

We write γ_i to denote the *prefix* of γ consisting of the first i messages. For readability, we often drop the input x from the notation, e.g. using $\ell = \ell(|x|)$, $\langle P, V\rangle = \langle P, V\rangle(x)$, etc.

The following two claims are shown in [AH, PT, GV]:

Claim 3.7 *For every* x,

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

Since $\langle P, V\rangle_{2i-1}$ is a prefix of $\langle P, V\rangle_{2i}$, the term $\mathrm{H}(\langle P, V\rangle_{2i}) - \mathrm{H}(\langle P, V\rangle_{2i-1})$ in the sum equals the conditional entropy $\mathrm{H}(\langle P, V\rangle_{2i} | \langle P, V\rangle_{2i-1})$. Thus, the sum measures the total entropy contributed by the verifier's messages, and it is natural that this should equal the number of coin tosses of the verifier. (Recall that the verifier reveals its coin tosses at the end.)

What is less obvious is that the sum should be significantly smaller when we consider the simulated transcripts for $x \in \Pi_N$.

Claim 3.8 *For every* $x \in \Pi_N$,

$$\sum_{i=1}^{\ell} [\mathrm{H}(S_{2i}) - \mathrm{H}(S_{2i-1})] \leq r - \log \frac{1}{s(|x|)} < r - 1.$$

Informally, this says that in case $x \in \Pi_N$ the simulated verifier is not behaving as randomly as the real verifier would — it captures at most at $s(|x|)$ fraction of the probability space of the verifer's messages. Intuitively, if this were not the case, then the simulator could be used to construct a prover strategy that convinces the verifier to accept with probability greater than $s(|x|)$, contradicting the soundness of the proof system.

Now, given input x, we construct circuits that sample from the following (joint) random variables.

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

When $x \in \Pi_Y$, then S is computationally indistinguishable from $\langle P, V\rangle$. So (X, Y) is indistinguishable from $(X', Y') = (\langle P, V\rangle_{2I}, \langle P, V\rangle_{2I-1})$, where I denotes a uniform random element of $\{1, \ldots, \ell\}$. By Claim 3.7, we have:

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

When $x \in \Pi_N$, then by Claim 3.7, we have

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

This is what we need to prove, except the entropy gap is only $1/\ell$. This can be increased to 1 by taking ℓ independent samples from the joint distribution. That is, we define $(\overline{X}, \overline{Y}) = ((X_1, \ldots, X_\ell), (Y_1, \ldots, Y_\ell))$, where the (X_i, Y_i)'s are independent copies of (X, Y). When $x \in \Pi_Y$, then $(\overline{X}, \overline{Y})$ is computationally indistinguishable from the analogously defined $(\overline{X'}, \overline{Y'})$, and $\mathrm{H}(\overline{X'}|\overline{Y'}) = \ell \cdot \mathrm{H}(X'|Y') = r$. And when $x \in \Pi_N$, then $\mathrm{H}(\overline{X}|\overline{Y}) = \ell \cdot \mathrm{H}(X|Y) \leq r - 1$.

Therefore the mapping $x \mapsto (\overline{X}, \overline{Y}), r$ satisfies Definition 3.4 ■

3.3 The SZK/OWF Characterization

In this section, we show that the Conditional Pseudoentropy Characterization implies the SZK/OWF Characterization.

Lemma 3.9 *If a promise problem satisfies the* Conditional Pseudoentropy Characterization*, then it also satisfies the* SZK/OWF Characterization.

The idea behind the proof is the following. If Π satisfies the Conditional Pseudoentropy Characterization, then on every YES instance, we obtain a samplable distribution (X, Y) that is computationally indistinguishable from (X', Y') where $\mathrm{H}(X'|Y')$ is large. We consider two cases. If, for the original distributions X, Y, we have that $\mathrm{H}(X|Y)$ is large, then the instance is information-theoretically distinguishable from a NO instance (where $\mathrm{H}(X|Y)$ is small), and such instances can be reduced to Entropy Difference, one of the complete problems for **SZK**. If instead $\mathrm{H}(X|Y)$ is small, then (X, Y) is computationally indistinguishable from a joint distribution with higher conditional entropy (namely (X', Y')). From such a pair, we can construct a one-way function using the techniques of Håstad, Impagliazzo, Levin, and Luby [HILL]. This case analysis provides the partition of YES instances into **SZK** instances and OWF instances.

Before proceeding with the actual proof, we state the result we need from [HILL].

Definition 3.10 *An* auxiliary-input false entropy generator on I *is a samplable auxiliary-input probability ensemble* $\mathcal{D} = \{D_x\}$ *for which there exists a samplable auxiliary-input probability ensemble* $\mathcal{F} = \{F_x\}$ *that is computationally indistinguishable from* $\mathcal{D}$ *on* I *and satisfies* $\mathrm{H}(F_x) \geq \mathrm{H}(D_x)+1$.

Note that the above definition refers to entropy, rather than conditional entropy as in the intuition above. We will need to cope with this in the proof. Also note that the definition requires that $\mathcal{F} = \{F_x\}$ is also samplable. This is actually not necessary (i.e. Lemma 3.11 below holds regardless), but we will achieve samplability of $\mathcal{F}$ in passing from conditional entropy to entropy, so we add include the samplability condition for consistency with [HILL].[9]

Lemma 3.11 ([HILL]) *If there exists an auxiliary-input false entropy generator on* I*, then there exists an auxiliary-input pseudorandom generator (and hence auxiliary-input one-way function) on* I*.*

Actually, since our goal at this point is only to construct a one-way function, some of the steps in the proof in the proof of Lemma 3.11 can be short-circuited. See the proof of Lemma 7.19, where review the details of this step (as part of generalizing our results to expected polynomial-time simulators).

Proof of Lemma 3.9: Given an instance x of the promise problem Π, we can efficiently construct two samplable distributions (X, Y) and parameter r such that if $x \in \Pi_Y$, then $\mathrm{H}(X'|Y') \geq r + 2$ for some (X', Y') indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 2$. (We may assume a gap of 4 without loss of generality by taking multiple independent samples from the joint distribution.)

Let I be the set of instances $x \in \Pi_Y$ such that $\mathrm{H}(X|Y) < r$. First we show that $\Pi' = (\Pi_Y \backslash I, \Pi_N)$ is in **SZK**. We prove this by reducing Π' to Entropy Difference. Consider the samplable distributions $A = (X, Y)$, $B = Y \otimes U_{r-1}$. Then

$$\mathrm{H}(A) - \mathrm{H}(B) = \mathrm{H}(X, Y) - (\mathrm{H}(Y) + (r - 1)) = \mathrm{H}(X|Y) - (r - 1).$$

Thus $\mathrm{H}(A) \geq \mathrm{H}(B) + 1$ when $x \in \Pi_Y \setminus I$, and $\mathrm{H}(B) \geq \mathrm{H}(A) + 1$ when $x \in \Pi_N$.

Now we show that we can construct a one-way function from instances in I. Note that when $x \in I$, we have $\mathrm{H}(X'|Y') \geq r + 2 > \mathrm{H}(X|Y) + 2$. Let $n = |x|$, let m be the number of bits output by X, set $k = 4n \cdot (m + n)^2$, and let $\mathcal{H}$ be an explicit family of universal hash functions mapping $\{0,1\}^{km}$ to $\{0,1\}^{kr+1}$. Let $s = O(km)$ be the number of random bits to choose a random hash function from $\mathcal{H}$. Consider the following samplable distribution

$$Z = (H, Y_1, \ldots, Y_k, H(X_1, \ldots, X_k)),$$

where H is a random hash function from $\mathcal{H}$, and the (X_i, Y_i)'s are independent copies of (X, Y). When $x \in I$, $\mathrm{H}(Z) \leq s + k \cdot \mathrm{H}(Y) + k \cdot r$. On the other hand, we will show below that that Z is computationally indistinguishable from the samplable distribution

$$Z' = (H, Y_1, \ldots, Y_k, U_{kr+1}),$$

[9] The samplability of $\mathcal{F}$ is only needed in [HILL] for proving results with respect to *uniform* adversaries. Indeed, the condition was not included in the conference version [ILL], which only dealt with nonuniform adversaries.

which has at least one more bit of entropy than Z. Thus, we have constructed an auxiliary-input false entropy generator on I, and thus by Lemma 3.11 there exists a one-way function on I, as desired.

We now proceed to show that when $x \in \Pi_Y$, Z is computationally indistinguishable from Z'. We know that there exist (X', Y') indistinguishable from (X, Y) such that $\mathrm{H}(X'|Y') \geq r + 2$. We can slightly modify (X', Y') to obtain (X^*, Y^*) indistinguishable from (X, Y) such that $\mathrm{H}(X^*|Y^*) \geq r + 1$ and $\Pr[X' = x|Y' = y] \geq 2^{-n} \cdot 2^{-m}$ for all $(x, y) \in \mathrm{Supp}(X', Y')$. (The pairs (x, y) for which the latter inequality does not hold constitute at most a 2^{-n} fraction of the probability mass under (X', Y'), and hence shifting their probability mass to other elements incurs a statistical difference of at most 2^{-n} and changes the conditional entropy by much less than 1 bit by Lemma 2.1.)

By a hybrid argument, Z is computationally indistinguishable from $Z^* = (H, Y_1^*, \ldots, Y_k^*, H(X_1^*, \ldots, X_k^*))$, where the (X_i^*, Y_i^*)'s are independent copies of (X^*, Y^*). By Lemma 2.5, $\overline{X^*} = (X_1^*, \ldots, X_k^*)$ is Δ-flat given $\overline{Y^*} = (Y_1^*, \ldots, Y_k^*)$ for $\Delta = \sqrt{k} \cdot (m + n)$. This implies that $(\overline{X^*}, \overline{Y^*})$ is 2^{-n}-close to some $(W, \overline{Y^*})$ such that for every $\overline{y} \in \mathrm{Supp}(\overline{Y^*})$, the min-entropy of W conditioned on $\overline{Y^*} = \overline{y}$ is at least

$$\begin{aligned} k \cdot \mathrm{H}(X^*|Y^*) - \sqrt{n} \cdot \Delta &\geq k \cdot (r + 1) - \sqrt{n} \cdot \Delta \\ &> kr + 2n + 1, \end{aligned}$$

where in the last inequality we use $\sqrt{n}\Delta \leq k/2$ and $2n + 1 \leq k/2$.

Thus, Z^* is statistically close to $(H, \overline{Y^*}, H(W))$, which is 2^{-n}-close to $(H, \overline{Y^*}, U_{kr+1})$ by the Leftover Hash Lemma (Lemma 2.6). This latter distribution is computationally indistinguishable from Z' because Y^* is computationally indistinguishable from Y. ■

3.4 The Indistinguishability Characterization

In this section, we show that the Indistinguishability Characterization is equivalent to the Conditional Pseudoentropy Characterization, and is thus satisfied by every problem in **HVZK**. This equivalence is proven using computational analogues of the reductions given in [Vad1, §3.4,§4.4] between the two complete problems for **SZK**, Statistical Difference and Entropy Difference.

Lemma 3.12 *If a promise problem satisfies the* Conditional Pseudoentropy Characterization*, then it satisfies the* Indistinguishability Characterization*.*

Proof: The reduction is identical to the one used in the proof of Lemma 3.9 to construct a pseudoentropy generator on the instances in I. Let Π be a promise problem satisfying the Conditional Pseudoentropy Characterization. As in the proof of Lemma 3.9, given an instance x of the promise problem Π, we can efficiently construct two samplable distributions (X, Y) and parameter r such that if $x \in \Pi_Y$, then $\mathrm{H}(X'|Y') \geq r + 2$ for some (X', Y') indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 2$. From X and Y, we can construct the samplable distributions Z and Z' as in the proof of Lemma 3.9. In that proof, it is shown that when $x \in \Pi_Y$, then Z and Z' are computationally indistinguishable. It is also shown that when $\mathrm{H}(X|Y) < r$ (in particular if $x \in \Pi_N$), then $\mathrm{H}(Z') \geq \mathrm{H}(Z) + 1$. By Lemma 2.1, this implies that $\Delta(Z, Z') \geq 1/2\ell$, where $\ell = \mathrm{poly}(n)$ is the number of bits output by Z and Z'. Applying Lemma 2.2, we can increase the statistical difference to $2/3$ on NO instances while maintaining computationally indistinguishability on YES instances. Thus, we conclude that Π satisfies the Indistinguishability Characterization. ■

Lemma 3.13 *If a promise problem satisfies the* Indistinguishability Characterization*, then it satisfies the* Conditional Pseudoentropy Characterization.

Proof: Let Π be a promise problem satisfying the Indistinguishability Characterization. Given an instance x of Π, we can efficiently construct two samplable distributions (X_0, X_1) such that X_0 and X_1 are computationally indistinguishable if $x \in \Pi_Y$ and $\Delta(X_0, X_1) \geq 2/3$ otherwise. Consider the following pair of jointly distributed random variables.

(B, Y)**:** Select $b \leftarrow \{0,1\}$. Sample $x \leftarrow X_b$. Output (b, x).

When $x \in \Pi_Y$, X_0 and X_1 are computationally indistinguishable. This implies that (B, Y) is computationally indistinguishable from (B', Y) where B' is a random bit independent of Y. Note that $\mathrm{H}(B'|Y) = 1$.

When $x \in \Pi_N$, $\Delta(X_0, X_1) \geq 2/3$. Intuitively, this means that B can be predicted with relatively high probability from $Y = X_B$ and hence B has less than 1 bit of entropy given Y. Specifically, it is shown in [Vad1, Claim 4.4.2] that $\mathrm{H}(B|Y) \leq \mathrm{H}_2((1+\delta)/2)$, where $\delta = \Delta(X_0, X_1)$. Plugging in $\delta = 2/3$, we see that $\mathrm{H}(B|Y) \leq \mathrm{H}_2(5/6) < .651$.

Thus the mapping $x \mapsto (B, Y), r = 1$ meets the requirements of the Conditional Pseudoentropy Characterization, except that the gap in conditional entropies between the two cases is only $1 - .651 = .349$ bits. The gap can be amplified to 1 bit by taking direct products as usual. ■

4 From the SZK/OWF Characterization to ZK

In this section, we construct a computational zero-knowledge proof system for every problem Π in **IP** that satisfies the SZK/OWF Characterization. Recall that the direct approach of using the **SZK** proof system on the **SZK** instances and the one-way-functions-based **IP**-to-**ZK** compiler on the OWF instances does not yield a zero-knowledge proof because the verifier learns whether the input is an **SZK** instance or a OWF instances. Thus, we take a more indirect approach, constructing "problem-dependent" commitment schemes for each of the two cases, then combining these commitments and using them for constructing a zero-knowledge proof.

4.1 Problem-Dependent Commitments

Roughly speaking, a *problem-dependent commitment scheme* is an auxiliary-input version of a commitment protocol, where the auxiliary input x (given to both the sender and receiver) is viewed as an instance of some promise problem Π. It is required that the scheme is hiding when $x \in \Pi_Y$ and is binding when $x \in \Pi_N$. 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.

An example, used in Bellare, Micali, and Ostrovsky [BMO], is based on the Graph Isomorphism problem: given graphs (G_0, G_1), a commitment to bit $b \in \{0,1\}$ is a random isomorphic copy of G_b. When $G_0 \cong G_1$, the commitment is perfectly hiding, and when $G_0 \ncong G_1$, then the commitment is perfectly binding. This idea was abstracted by Itoh, Ohta, and Shizuya [IOS], who studied the general utility of language-dependent commitment schemes for constructing zero-knowledge proofs. Specifically, they showed that every language possessing a noninteractive language-dependent commitment scheme that is perfectly binding and perfectly hiding is in **PZK**, as is the complement of every such language. Recently, in [MV], the notion was

further generalized to allow interactive commitments, statistical security, and promise problems, and was suggested as a possible tool for proving that every problem in $\mathbf{SZK} \cap \mathbf{NP}$ has a statistical zero-knowledge proof system with an efficient prover.

Here we consider further relaxations of the definition — we allow the sender's algorithm to be computationally unbounded (rendering it useless for the application in [MV]), allow the hiding property to be computational, and only require security for an honest receiver (i.e. one that follows the specified protocol). The fact that the sender is not polynomial time complicates the definition substantially, because many commonly used properties of commitment schemes implicitly use the fact that the sender algorithm is polynomial time. For example, standard commitment schemes are "zero knowledge" in the sense that the receiver learns nothing other than the bit to which the sender commits; this is the case because the receiver can simulate a commitment to bit b by simply running the sender's algorithm. Instead, we will need to explicitly include such properties in the definition.

Definition 4.1 *For a promise problem* Π*, an (unbounded-sender, honest-receiver)* Π-dependent commitment scheme *consists of two interactive protocols* (S_1, R_1) *(the* commitment phase*) and* (S_2, R_2) *(the* reveal phase*) and a promise problem* $\text{VAL} = (\text{VAL}_Y, \text{VAL}_N)$*. In the commitment phase, both* S_1 *and* R_1 *receive a common input* $x \in \{0,1\}^*$*,* S_1 *receives a private input* $b \in \{0,1\}$*, and the protocol produces as output a* commitment z*. In the reveal phase, both* S_2 *and* R_2 *receive the common input* $x \in \{0,1\}^*$*, a commitment* z*, and a bit* $b \in \{0,1\}$*, and at the end of the protocol,* R_2 *accepts or rejects. We write* $(S_1(b), R_1)(x)$*,* $(S_2, R_2)(x, z, b)$*, and* $(S, R)(x, b)$ *to denote the interaction between* S *and* R *in the commit phase, reveal phase, and the two phases combined, respectively.*

We require the following conditions:

1. *(Efficiency)* $R = (R_1, R_2)$ *is computable in probabilistic polynomial time (in the length of the common input* x*). (*S *is allowed to be computationally unbounded.)*

2. *(Completeness) For all* $x \in \{0,1\}^n$ *and all* $b \in \{0,1\}$*, if we let* z *be the output of* $(S_1(b), R_1)(x)$*, then* $(x, z, b) \in \text{VAL}_Y$ *with probability* $1 - \text{neg}(n)$*.*

3. *(Validity Tests)* (S_2, R_2) *is an interactive proof system (with negligible error probabilities) for* VAL*. Moreover, the promise problem* VAL *is in* $\mathbf{AM}$*.*

4. *(Zero Knowledge) There is a probabilistic polynomial-time algorithm* M *such that for every* $x \in \{0,1\}^*$ *and* $b \in \{0,1\}$*, the distribution* $M(x, b)$ *has statistical difference* $\text{neg}(n)$ *from* R*'s view of* $(S, R)(x, b)$*.*

5. *(Computationally hiding on* YES *instances) If* $x \in \Pi_Y$*, then* R*'s views in* $(S_1(0), R_1)(x)$ *and* $(S_1(1), R_1)(x)$ *are computationally indistinguishable. In case these views are statistically indistinguishable, we will refer to the scheme as* statistically hiding*.*

6. *(Statistically binding on* NO *instances) If* $x \in \Pi_N$*, then for every* S^**, if we let* z *be the output of* $(S_1^*, R_1)(x)$*, then with probability at least* $1 - \text{neg}(n)$*, either* $(x, z, 0)$ *or* $(x, z, 1)$ *is in* VAL_N*.*

A few remarks on the above conditions:

- The fact that we allow S to be computationally unbounded results in several differences between the above definition and standard definitions of commitment schemes. When S is restricted to be polynomial time, the zero-knowledge condition is trivial to satisfy (because $M(x,b)$ could carry out an execution of $(S,R)(x,b)$) and thus is typically omitted, and the reveal phase can wlog consist of S just sending its coin tosses to R. On the other hand, we do not need S to retain state between the two phases (because it can generate a random state consistent with the commitment z).

- The completeness and zero-knowledge conditions (and the validity tests) are required for all inputs $x \in \{0,1\}^*$, not just those that satisfy the promise of Π. This will be useful in combining two problem-dependent commitment schemes to obtain one for the union of the corresponding promise problems.

- The definition provides for two different kinds of validity tests. One is the specified protocol (S_2, R_2) (which may have many rounds, but is "zero knowledge" according to Item 4). The other is the (unspecified) **AM** protocol for VAL (which has only two rounds). Both will be useful for us.

- Both the zero knowledge and hiding conditions are only required for honest (but curious) receivers. The result is that the proof systems we construct using such commitments will only be honest-verifier zero knowledge. We will then obtain zero knowledge against cheating verifier strategies using the compiler of [GSV1].

Our results will show that problem-dependent commitment schemes characterize **ZK** and **SZK**:

Theorem 4.2 $\Pi \in$ **ZK** *if and only if* Π *has a computationally hiding, public-coin problem-dependent commitment scheme in the sense of Definition 4.1.*

Theorem 4.3 $\Pi \in$ **SZK** *if and only if* Π *has a statistically hiding problem-dependent commitment scheme in the sense of Definition 4.1.*

These theorems demonstrate that commitment schemes are at the heart of all zero-knowledge proofs.

In this section, however, we prove just the following:

Lemma 4.4 *If a promise problem* Π *satisfies the* SZK/OWF CHARACTERIZATION*, then there exists a public-coin* Π*-dependent commitment scheme (in the sense of Definition 4.1). Moreover the sender can be implemented in probabilistic polynomial time with an* **NP** *oracle.*

We will prove this by dealing separately with the **SZK** instances and OWF instances. The OWF instances are a straightforward application of the known construction of commitment schemes from one-way functions.

Lemma 4.5 *If there exists an auxiliary-input one-way function on set* I*, then there is a* Π*-dependent commitment scheme for the promise problem* $\Pi = (I, \overline{I})$*. Moreover, this commitment scheme is public coin and the sender can be implemented in probabilistic polynomial time.*

Proof: By Theorem 2.14, we can construct an auxiliary-input pseudorandom generator $G_x : \{0,1\}^{p(|x|} \to \{0,1\}^{3p(|x|)}$ on I. Now we use Naor's commitment scheme from pseudorandom generators [Nao]:

Commit Phase $(S_1(b), R_1)(x)$**,** where $|x| = n$.

1. R_1 chooses $v \leftarrow \{0,1\}^{3p(n)}$ and sends v to S_1.
2. S_1 chooses $r \leftarrow \{0,1\}^{p(n)}$ and $w = G_x(r) \oplus bv$ to R_1.
3. The commitment z is defined to be the pair (v, w).

Now we define the promise problem $\textsc{Val} = (\textsc{Val}_Y, \textsc{Val}_N)$ by

$$\begin{aligned} \textsc{Val}_Y &= \{(x,(v,w),b) : \exists r \in \{0,1\}^{p(|x|)}\ w = G(x,r) \oplus bv\}, \text{ and} \\ \textsc{Val}_N &= \overline{\textsc{Val}_Y}. \end{aligned}$$

Clearly $\textsc{Val} \in \mathbf{NP}$, and in fact the reveal phase simply consists of the sender providing the standard **NP** proof that $(x,(s,v),b) \in \textsc{Val}_Y$ (namely r such that $w = G_x(r) \oplus bv$).

The completeness and public coin properties hold by inspection. The zero knowledge condition holds because the sender is polynomial time. Following [Nao], the (computational) hiding property on $x \in I$ follows from the pseudorandomness of G_x on such instances. Specifically, we know that $G_x(U_{p(n)})$ is indistinguishable from $U_{3p(n)}$. Thus, if we let the random variable $V \leftarrow \{0,1\}^{3p(n)}$ denote the message of R_1, we see that $(V, G_x(U_{p(n)}) \oplus V)$ is indistinguishable from $(V, U_{3p(n)} \oplus S) = (V, U_{3p(n)})$, and note that the former distribution is R_1's view of a commitment to 1 and the latter is R_1's view of a commitment to 0. Following [Nao], the (statistical) binding property on $x \notin I$ (in fact on all $x \in \{0,1\}^*$) follows from the fact that G_x is length-tripling. Specifically, with probability at least $1 - 2^{-p(n)}$ over $v \leftarrow \{0,1\}^{3p(n)}$, the image of G_x will be disjoint from the image of $G \oplus v$, in which case there is no w such that (v, w) is a valid commitment of both 0 and 1. ■

For the **SZK** instances, we prove the following (which is the forward direction of Theorem 4.3) in Section 5.

Lemma 4.6 *Every problem* Π *in* **SZK** *has a* Π*-dependent commitment scheme. Moreover, the scheme is public coin and statistically hiding, and the sender can be implemented in probabilistic polynomial time with an* **NP** *oracle.*

We now show how to combine these two commitment schemes with to prove Lemma 4.4.

Lemma 4.7 *If promise problems* $\Pi = (\Pi_Y, \Pi_N)$ *and* $\Gamma = (\Gamma_Y, \Gamma_N)$ *each have problem-dependent commitment schemes, then the promise problem* $\Pi \cup \Gamma \stackrel{\text{def}}{=} (\Pi_Y \cup \Gamma_Y, \Pi_N \cap \Gamma_N)$ *has a problem-dependent commitment scheme.*

Proof: Let (S', R') be the problem-dependent commitment scheme for Π, and (S'', R'') the one for Γ, with valid commitments defined by promise problems $\textsc{Val}'$ and $\textsc{Val}''$. Intuitively, on an input x, we would like to use (S', R') if $x \in \Pi_Y$ and use (S'', R'') if $x \in \Gamma_Y$. Unfortunately, we do not know which is the case. So we will use *both*, and do so in such a way that the resulting scheme is hiding even when only one of the two is hiding.

Specifically the new scheme $(S, R) = ((S_1, S_2), (R_1, R_2))$ is constructed as follows:

Commit Phase $(S_1(b), R_1)(x)$

1. S_1 chooses random $b', b'' \leftarrow \{0,1\}$ such that $b' \oplus b'' = b$.
2. S_1 and R_1 execute $(S_1'(b'), R_1')(x)$ and $(S_1''(b''), R_1'')(x)$ to obtain a commitments z',z'', respectively.

3. The output commitment is $z = (z', z'')$.

Valid Commitments The promise problem of valid commitments is defined to be $\text{VAL} = (\text{VAL}_Y, \text{VAL}_N)$ where

$$\begin{aligned}\text{VAL}_Y &= \{(x,(z',z''),b) : \exists b', b'' \in \{0,1\}\ [b' \oplus b'' = b] \wedge [(x,z',b') \in \text{VAL}'_Y] \wedge [(x,z'',b'') \in \text{VAL}''_Y]\} \\ \text{VAL}_N &= \{(x,(z',z''),b) : \forall b', b'' \in \{0,1\}\ [b' \oplus b'' \neq b] \vee [(x,z',b') \in \text{VAL}'_N] \vee [(x,z'',b'') \in \text{VAL}''_N]\}\end{aligned}$$

Reveal Phase $(S_2, R_2)(x,(z',z''),b)$**:** 1. S_2 sends b', b'' such that $b' \oplus b'' = b$.

2. R_2 checks that $b' \oplus b'' = b$ and rejects immediately if not.
3. S_2 and R_2 execute $(S', R')(x, z', b')$ and $(S'', R'')(x, z'', b'')$, and R_2 accepts if both R' and R'' accept.

The completeness property of (S, R) on all x follows from the completeness properties of (S', R') and (S'', R''), which guarantee that whp $(x, z', b') \in \text{VAL}'_Y$ and $(x, z'', b'') \in \text{VAL}''_Y$, and hence $(x,(z',z''),b) \in \text{VAL}_Y$. (Here it is important that we required completeness to hold on all instances, rather than just YES instances, since Π_Y and Γ_Y need not be the same.) The zero-knowledge property follows in a similar manner: The new simulator $M(x,b)$ chooses $b', b'' \leftarrow \{0,1\}$ such that $b' \oplus b'' = b$, runs the original simulators $M'(x,b')$ and $M''(x,b'')$, and combines their outputs to simulate the view of R. The fact that (S_2, R_2) is a an interactive proof for VAL follows by inspection.

For the hiding property on $\Pi_Y \cup \Gamma_Y$, suppose wlog that $x \in \Gamma_Y$. Note that the view of R_1 in $(S_1(b), R_1)(x)$ consists of the view of R'_1 in $(S'_1(b'), R'_1)(x)$ concatenated with the view of R''_1 in $(S''_1(b''), R''_1)(x)$, where b' and b'' are chosen randomly such that $b' \oplus b'' = b$. The first part of the view (namely the R'_1-view) has the same distribution regardless of the value b, because b' is a random bit. Thus it suffices to show that for every fixed value of b' and the R'_1-view, the R''_1-view in case $b'' = b'$ (i.e., $b = 0$) is indistinguishable from the R''_1-view in case $b'' \neq b'$ (i.e., $b = 1$). But this follows from the hiding property of (S'', R'') on $x \in \Gamma_Y$.

The binding property on $x \in \Pi_N \cap \Gamma_N$ follows from the binding properties of the two commitment schemes: For every strategy S^*, we know that with high probability, the output (z', z'') of (S^*, R) satisfies the following. There is at most one $b' \in \{0,1\}$ such that $(x, z', b') \notin \text{VAL}'_N$ and there exists at most one $b'' \in \{0,1\}$ such that $(x, z'', b'') \notin \text{VAL}''_N$. Thus there is at most one b (namely $b = b' \oplus b''$) such that $(x,(z',z''),b) \notin \text{VAL}_N$, as desired. ■

4.2 The Zero-Knowledge Proof

Lemma 4.8 *If a promise problem Π is in* **IP** *and has a computationally hiding (resp., statistically hiding), public-coin problem-dependent commitment scheme (in the sense of Definition 4.1), the $\Pi \in$* **HVZK** *(resp., $\Pi \in$* **HVSZK** $=$ **SZK***). Moreover, the honest-verifier zero-knowledge proof for Π is public coin and the prover's strategy P'_x on can be computed in probabilistic polynomial time with oracles for S_x and P_x, where S is the sender algorithm in the problem-dependent commitment scheme where P is a prover in any public-coin interactive proof system for Π.*

Proof: We begin with the special case that $\Pi \in$ **NP**, where we use the ideas of Itoh, Ohta, and Shizuya [IOS] with our more general notion of problem-dependent commitments. The approach is to use the zero-knowledge proofs of Goldreich, Micali, and Wigderson for all of **NP**, replacing the commitment scheme used there with the problem-dependent commitment for Π. An outline of the steps of the resulting protocol follows.

Zero-knowledge proof $(P, V)(x)$**:**

1. Both parties reduce x to an instance G of Three-Coloring.
2. P selects a random 3-coloring C of G.
3. P commits to the coloring by engaging with V in (polynomially many executions of) the commitment phase of Π-dependent commitment scheme.
4. V selects a random edge e in G.
5. P reveals the colorings of the endpoints of e, and proves their validity to V via the reveal phase of the Π-dependent commitment scheme.
6. V accepts if it the colors of the endpoints are different and it accepted in both executions of the reveal phase.

Completeness when $x \in \Pi_Y$ follows from completeness of the Π-dependent commitment scheme. Soundness when $x \in \Pi_N$ follows from the binding property of the Π-dependent commitment scheme when $x \in \Pi_N$. (Honest-verifier) zero knowledge follows from the hiding and zero-knowledge properties of the commitment scheme. Specifically, the simulator chooses a random edge e in the graph (to be the verifier's challenge), chooses two random distinct colors for its endpoints, and arbitrarily extends this to a coloring of the entire graph. It uses the simulator for the commitment scheme to simulate all the commitments, using the simulated commitment phase for all the commitments, but the simulated reveal phase only for the edge e. The hiding property of the commitment scheme implies that this simulation is computationally indistinguishable from the (honest) verifier's view.

For the general case that $\Pi \in \mathbf{IP}$, we follow [IY, BGG$^+$] and transform an interactive proof (P, V) for Π into a zero-knowledge proof. By [GS], we may assume that (P, V) is public coin. An outline of the zero-knowledge proof follows:

Zero-knowledge proof $(P', V')(x)$**:**

1. (P', V') simulate the public-coin interactive proof $(P, V)(x)$, but instead of sending P's messages explicitly, P' commits to P's messages using the commit phase of the Π-dependent commitment scheme. (The public-coin nature of (P, V) ensures that V can compute its messages without seeing P's messages explicitly.) Let $(z_1, \ldots, z_m)$ be all the commitments obtained in this way.
2. V sends a random strings $r_1, \ldots, r_m$ for the **AM** proof system for Val.
3. Now P proves the following **NP** statement to V using protocol described above (GMW+problem-dependent commitments): there exist decommitments $b_1, \ldots, b_m$ such that (a) V would have accepted in the interactive proof if the prover responses were given by $b_1, \ldots, b_m$, and (b) there are prover responses $s_1, \ldots, s_m$ such that the **AM** verifier for Val would accept on transcript $((x, z_i, b_i), r_i, s_i)$ for $i = 1, \ldots, m$.

The analysis of this proof system is similar to the previous one. The claim about the prover complexity follows by inspection. ■

We see that the characterization of **SZK** in terms of problem-dependent commitments (Theorem 4.3 follows from Lemma 4.6 (to be proved in Section 5) and Lemma 4.8.

The above gives honest-verifier zero-knowledge proofs. These can be converted to zero-knowledge proofs that tolerate cheating verifiers using the following compiler of Goldreich, Sahai, and Vadhan [GSV1].

Theorem 4.9 ([GSV1]) *Any honest-verifier public-coin statistical (resp., computational) zero-knowledge proof system can be transformed into a (cheating-verifier) public-coin statistical (resp., computational) zero-knowledge proof system. Furthermore,*

1. *The resulting proof system has twice as many rounds as the original one.*
2. *The resulting prover strategy can be implemented in probabilistic polynomial time given oracle access to the original prover strategy.*
3. *The resulting proof system has completeness error $2^{-\Omega(n)}$ and soundness error $1/k$, where k is the security parameter. In case the original proof system has perfect completeness, so does the resulting one.*
4. *The resulting proof system has a black-box simulator.*

5 Problem-dependent commitments for SZK.

In this section, we construct our problem-dependent commitment schemes for **SZK**, thereby proving Lemma 4.6. This is the technically most involved part of our work.

5.1 Overview

We will construct a problem-dependent commitment scheme for the **SZK**-complete problem Statistical Difference [SV]. This means that we will design a commitment protocol in which both the sender and receiver get as auxiliary input a pair (X_0, X_1) of samplable distributions. The commitment scheme should be (statistically) hiding when X_0 and X_1 are statistically close and (statistically) binding when X_0 and X_1 are statistically far apart. By the Polarization Lemma of [SV], we may assume w.l.o.g. that the statistical difference between X_0 and X_1 is either exponentially small (for yes instances) or exponentially close to 1 (for no instances).

A natural idea, suggested in [MV], is the following. To commit to a bit b, the sender sends a random sample $x \leftarrow X_b$. To decommit, the sender reveals b and the coin tosses r used to generate the sample, and the receiver verifies that $x = X_b(r)$.

When X_0 and X_1 are statistically close, this scheme is indeed hiding. When $\Delta(X_0, X_1) = 1$ (i.e. X_0 and X_1 have disjoint supports), then the scheme is perfectly binding. But we are only guaranteed that $\Delta(X_0, X_1)$ is exponentially close to 1, and this does not suffice for any sort of hiding. Indeed, two distributions can have statistical difference exponentially close to 1 and yet have identical supports (which means that every commitment can be opened in two ways).

To deal with this problem, we notice that the intersection between the supports can consist of two kinds of elements. First, there can be samples that are atypically light for at least one of the distributions (i.e. have probability mass much smaller than 2^{-h}, if we assume (wlog) that $\mathrm{H}(X_0) = \mathrm{H}(X_1) = h$). There can be very many such elements. Second, there can be samples

that are not atypically light for either distribution. However, there can only be a small number of elements ($\ll 2^h$) of this type, if the distributions have statistical difference exponentially close to 1. Still, we need to cope with both kinds of samples.

To deal with the latter problem, we replace the commit phase with an interactive protocol whereby the receiver constrains the sender's choice of the sample/commitment x. Even if the sender deviates from the protocol, with high probability the commit phase will produce a sample that is atypically light for at least one of the two distributions, in which case we will regard it as a commitment to the bit corresponding to the *other* distribution. Thus, to reveal a commitment to bit b, the sender will give an (interactive) proof that the sample is not atypically light for X_b. Of course, the challenge is to design both of these protocols so that the hiding property is maintained in case of YES instances.

Fortunately, there are two protocols due to Okamoto [Oka] (see also [GV, Vad1]) that turn out to be very well-suited for these tasks. We use an adaptation of Okamoto's "Sample Generation Protocol" for the commitment phase, and his "Sample Test Protocol" for the reveal phase. The price we pay for using Okamoto's protocols is that the sender can no longer be implemented in probabilistic polynomial time (but rather $\mathbf{BPP}^{\mathbf{NP}}$), and also that the round complexity becomes polynomial rather than constant.

5.2 Preprocessing the Distributions

We will not apply Okamoto's protocols directly to instances of STATISTICAL DIFFERENCE itself, but rather do some preprocessing on the distributions. The first drives the thresholds α, β exponentially close to 0 and 1, respectively.

Lemma 5.1 (Polarization Lemma [SV]) *There is a polynomial-time computable function mapping pairs of distributions (X_0, X_1) (specified by circuits which sample from them) and a unary parameter 1^k to pairs of distributions (Y_0, Y_1) such that:*

$$\begin{aligned} \Delta(X_0, X_1) \geq 2/3 &\Rightarrow \Delta(Y_0, Y_1) \leq 2^{-k} \\ \Delta(X_0, X_1) \leq 1/3 &\Rightarrow \Delta(Y_0, Y_1) \geq 1 - 2^{-k}, \end{aligned}$$

The second transformation we will use is simply taking Direct Products, as analyzed in Section 2.2. Combining these two transformations, we prove:

Lemma 5.2 *For every promise problem $\Pi \in \mathbf{SZK}$, there is a polynomial-time computable function mapping instances x of length n and unary parameters 1^k, 1^ℓ to pairs of distributions (Z_0, Z_1) such that:*

- *If $x \in \Pi_Y$, then $\Delta(Z_0, Z_1) \leq \ell \cdot 2^{-k}$.*
- *If $x \in \Pi_N$, then $\Delta(Z_0, Z_1) \geq 1 - 2^{-\ell}$.*
- *For all x, $\mathrm{H}(Z_0) = \mathrm{H}(Z_1)$ and both Z_0 and Z_1 are $\sqrt{\ell} \cdot \mathrm{poly}(n, k)$-flat.*

When we apply this lemma, we will set $k = O(n)$, and $\ell \gg k$. The key point for us is that the statistical difference in the case of NO instances goes to 1 exponentially fast with ℓ, whereas the deviation from flatness grows sublinearly with ℓ. We will show that this implies that the intersection of the supports of the two distribution is due only to (a) atypically light elements, and (b) a *small* number of other elements (i.e. much fewer than $2^{\mathrm{H}(Z_b)}$).

Proof: Let an instance x of $\Pi \in \mathbf{SZK}$ and the parameters $1^k, 1^\ell$ be given. By the completeness of Statistical Difference and the Polarization Lemma (Lemma 5.1), we can produce in polynomial time distributions (Y_0, Y_1) such that

$$\begin{aligned} x \in \Pi_Y &\Rightarrow \Delta(Y_0, Y_1) \leq 2^{-2k} \\ x \in \Pi_N &\Rightarrow \Delta(Y_0, Y_1) \geq 1 - 2^{-2k} \geq 1/2, \end{aligned}$$

Now, let $W_0 = Y_0 \otimes Y_1$ (i.e. a sample of Y_0 followed by an independent sample of Y_1) and $W_1 = Y_1 \otimes Y_0$. This ensures $\mathrm{H}(W_0) = \mathrm{H}(W_1)$, and

$$\begin{aligned} x \in \Pi_Y &\Rightarrow \Delta(W_0, W_1) \leq 2 \cdot 2^{-2k} \\ x \in \Pi_N &\Rightarrow \Delta(W_0, W_1) \geq 1/2, \end{aligned}$$

Now we let $Z_0 = \otimes^{c \cdot \ell} W_0$ and $Z_1 = \otimes^{c \cdot \ell} W_1$, for a sufficiently large constant c. Then, by the Lemma 2.2,

$$\begin{aligned} x \in \Pi_Y &\Rightarrow \Delta(Z_0, Z_1) \leq c\ell \cdot 2 \cdot 2^{-2k} \leq \ell \cdot 2^{-k} \\ x \in \Pi_N &\Rightarrow \Delta(Z_0, Z_1) \geq 1 - \exp(-\Omega(c\ell)) \geq 1 - 2^{-\ell}, \end{aligned}$$

for an appropriate constant c and sufficiently large k. Also $\mathrm{H}(Z_0) = c\ell \cdot \mathrm{H}(W_0) = \mathrm{H}(Z_1)$. And if $m = \mathrm{poly}(n, k)$ is the number of input gates to W_0 and W_1, then $\Pr[W_b = w] \geq 2^{-m}$ for all $b \in \{0, 1\}$ and all w in the support of W_b, so the Flattening Lemma tells us that Z_0 and Z_1 are both $\sqrt{\ell} \cdot m$-flat. ■

The following lemma shows that for two nearly flat distributions with statistical difference very close to 1, there can only be a small number of strings that are typical or heavy for both distributions.

Lemma 5.3 *Suppose Z_0 and Z_1 are random variables such that $\mathrm{H}(Z_0) = \mathrm{H}(Z_1)$ and $\Delta(\mathrm{H}(Z_0), \mathrm{H}(Z_1)) \geq 1 - 2^{-\ell}$. Then for any $\Delta > 0$,*

$$\#\{z : z \text{ is not } \Delta\text{-light for } Z_0 \text{ and } z \text{ is not } \Delta\text{-light for } Z_1\} \leq \frac{2^{H(Z_0)}}{2^{\ell - \Delta}}.$$

Proof: Let S be the set of z that are neither Δ-light for Z_0 nor for Z_1. Then

$$\begin{aligned} 2^{-\ell} &\geq 1 - \Delta(Z_0, Z_1) \\ &= \sum_z \min\{\Pr[Z_0 = z], \Pr[Z_1 = z]\} \\ &> \sum_{z \in S} \min\left\{2^{-\Delta} \cdot 2^{-\mathrm{H}(Z_0)}, 2^{-\Delta} \cdot 2^{-\mathrm{H}(Z_1)}\right\} \\ &= |S| \cdot 2^{-\Delta} \cdot 2^{-\mathrm{H}(Z_0)}. \end{aligned}$$

Thus, $|S| < 2^{\mathrm{H}(Z_0)}/2^{\ell - \Delta}$, as desired. ■

Applying this to the (Z_0, Z_1) constructed in Lemma 5.2, where Z_0, Z_1 are Δ-flat for $\Delta \ll \ell$, we see that on NO instances, there are very few ($\ll 2^{\mathrm{H}(Z_b)}$) elements in the intersections of the supports, except those that are atypically light.

5.3 Okamoto's subprotocols

We now describe the two protocols of Okamoto that we will use in our commitment scheme. The first is used for generating a random sample from a nearly flat distribution so that even if one party cheats, the output will be unlikely to fall in a sufficiently small set. The second is used to test that a sample from a nearly flat distribution is not too light. Our presentation of these protocols follows [GV, Vad1] (though we call the parties S and R instead of M and A for consistency with the definition of problem-dependent commitments).

Below, all distributions are given in the form of a circuit which generate them. The input to these protocols will consist of a distribution, denoted X. We will denote by m (resp., n) the length of the input to (resp., output of) the circuit generating the distribution X. In order to define the notion of a sample generation protocol, we must formalize what it means for an interactive protocol to have output.

Definition 5.4 (sample generation protocol) *A protocol (S, R) is called a* sample generation protocol *if on common input a distribution X and parameters Δ, t, such that X is Δ-flat and $1 \leq t \leq \Delta$, the following holds:*

1. *(Efficiency) R is computable in probabilistic polynomial time, and S is computable in probabilistic polynomial time with an* **NP** *oracle.*
2. *("Completeness") If both parties are honest, then the output of the protocol has statistical difference at most $m \cdot 2^{-\Omega(t^2)}$ from X.*
3. *("Soundness I") If R is honest then, no matter how S plays, the output will be $2\sqrt{t\Delta}\cdot\Delta$-heavy with probability at most $m \cdot 2^{-\Omega(t^2)}$.*
4. *("Soundness II") If R is honest then for every set $T \subseteq \{0,1\}^n$ of size at most $2^{-6\sqrt{t\Delta}\cdot\Delta} \cdot 2^{\mathrm{H}(X)}$, no matter how S plays, the output will be in T with probability at most $m \cdot 2^{-\Omega(t^2)}$.*
5. *(Strong "Zero Knowledge") There exists a probabilistic polynomial-time* simulator *M so that for every (X, Δ, t) as above, the following two distributions have statistical difference at most $m \cdot 2^{-\Omega(t^2)}$:*
   - **(A)** *Execute (S, R) on common input (X, Δ, t) and output the view of R, appended by the output.*
   - **(B)** *Choose $x \leftarrow X$ and output $(M(X, \Delta, t, x), x)$.*

A sample generation protocol is said to be public coin *if it is public coin for R.*

In [Oka, GV, Vad1], only the first soundness condition is given, but we will actually use the second. (But our proof that the protocol satisfies the second will make use of the first.) The above zero-knowledge property is referred to as *strong* since the simulator cannot produce a view-output pair by first generating the view and then computing the corresponding output. Instead, the simulator is forced (by the explicit inclusion of x in Distribution (B)) to generate a consistent random view for a given random output (of the protocol). We comment that the trivial protocol in which R uniformly selects an input r to the circuit X and reveals both r and the output $x = X(r)$ cannot be used since the simulator is only given x and it may be difficult to find an r yielding x in

general. Still, a sample generation protocol is implicit in Okamoto's work [Oka] (where it is called "pre-test'); see Protocol 5.6. Note also that the zero-knowledge condition implies the completeness condition; still conceptually it is convenient to state them separately.

Theorem 5.5 (implicit in [Oka], explicit in [GV]) *There exists a public-coin sample generation protocol. Furthermore, the number of messages exchanged in the protocol is linear in m.*

Protocol 5.6: Sample generation protocol (S, R)

Input: (X, Δ, t), where $t \leq \Delta$

1. S: Select $x_0 \in \{0,1\}^n$ according to X and send x_0 to R.
2. S, R: Repeat for i from 1 to m:
   (a) R: Choose h_i uniformly from a family of pairwise independent hash functions mapping $\{0,1\}^{m+n}$ to $\{0,1\}^{m-3t\Delta}$ and send h_i to S.
   (b) S: Choose (r_{i-1}, x_i) from the distribution $\{r : X(r) = x_{i-1}\} \times X$, conditioned on $h(r_{i-1}, x_i) = 0$, and send (r_{i-1}, x_i) to R. (If there is no such pair (r, x'), then S sends `fail` to R.)
   (c) R: Check that $X(r_{i-1}) = x_{i-1}$ and $h(r_{i-1}, x_i) = 0$. If either condition fails, reject.

Output: x_m, unless R rejects in some iteration of the above loop, in which case output any canonical string outside $\{0,1\}^n$, e.g. 0^{n+1}.

In [Oka, GV], it is proven that Protocol 5.6 satisfies all of the properties in Definition 5.4, except the sender complexity and Soundness II. The sender complexity follows from the observation that the sender only needs to sample strings uniformly from efficiently decidable sets (i.e. satisfying assignments to a known, polynomial-sized circuit), and it is known how to do such sampling given an **NP** oracle [JVV, BGP].

Lemma 5.7 *Protocol 5.6 satisfies the Soundness II condition of Definition 5.4.*

Proof: Fix a set T of size at most $2^{-6\sqrt{t\Delta}\cdot\Delta} \cdot 2^{\mathrm{H}(X)}$. We need to show that the output x_m is in T with probability at most $m \cdot 2^{-\Omega(t^2)}$, even under a cheating strategy for S. The Soundness I condition says that x_m is $2\sqrt{t\Delta} \cdot \Delta$-heavy with probability at most $m \cdot 2^{-\Omega(t^2)}$. In fact, the proof of this condition [GV] also shows that x_{m-1} is $2\sqrt{t\Delta} \cdot \Delta$-heavy with probability at most $m \cdot 2^{-\Omega(t^2)}$. (Indeed, the protocol could have been terminated after $m-1$ or even slightly fewer stages, but m was chosen as a clean upper-bound on the number of stages needed.) We will show that if x_{m-1} is not $2\sqrt{t\Delta} \cdot \Delta$-heavy, then the probability (over h_m) that S can select x_m to be in T (without R rejecting) is at most $2^{-\Omega(t^2)}$.

The number N of strings r_{m-1} such that $X(r_{m-1}) = x_{m-1}$ is

$$N = 2^m \cdot \Pr[X = x_{m-1}] < 2^m \cdot 2^{2\sqrt{t\Delta}\cdot\Delta} \cdot 2^{-\mathrm{H}(X)}.$$

Thus, the number of pairs (r_{m-1}, x_m) such that $X(r_{m-1}) = x_{m-1}$ and $x_m \in T$ equals

$$N \cdot |T| = \left(2^m \cdot 2^{2\sqrt{t\Delta}\cdot\Delta} \cdot 2^{-\mathrm{H}(X)}\right) \cdot \left(2^{-6\sqrt{t\Delta}\cdot\Delta} \cdot 2^{\mathrm{H}(X)}\right) \leq 2^{-t^2} \cdot 2^{m-3t\Delta},$$

where the last inequality uses $t \leq \Delta$. Since $h_m(z)$ is uniformly distributed in $\{0,1\}^{m-3t\Delta}$ for every z, the probability that there exists a pair (r_{m-1}, x_{m-1}) such that $h_m(r_{m-1}, x_{m-1}) = 0$ is at most 2^{-t^2}. ■

The second protocol tests whether a sample is too light; here we do not need any modifications from the definition in [GV].

Definition 5.8 (sample test protocol) *A protocol (S, R) is called a* sample test protocol *if on common input a distribution X, a string $x \in \{0,1\}^n$ and parameters Δ, t, such that X is Δ-flat and $t \leq \Delta$, the following holds:*

1. *(Efficiency) R is computable in probabilistic polynomial time, and S is computable in probabilistic polynomial time with an* **NP** *oracle.*
2. *("Completeness") If both parties are honest and x is $t\cdot\Delta$-typical then R accepts with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$.*
3. *("Soundness") If x is $6\sqrt{t\Delta}\cdot\Delta$-light and R is honest then, no matter how S plays, R accepts with probability at most $m \cdot 2^{-\Omega(t^2)}$.*
4. *(Weak "Zero Knowledge") There exists a probabilistic polynomial-time* simulator *M so that for every (X, Δ, t) as above and for every $t \cdot \Delta$-typical x, the following two distributions have statistical difference at most $m \cdot 2^{-\Omega(t^2)}$:*
   - **(A)** *Execute (S, R) on common input (X, x, Δ, t) and output the view of R, prepended by x.*
   - **(B)** *Choose r uniformly in $\Omega_X(x) \stackrel{\text{def}}{=} \{r' : X(r') = x\}$, and output $(x, M(X, x, \Delta, t, r))$.*

A sample test protocol is said to be public coin *if it is public coin for R.*

The above zero-knowledge property is referred to as *weak* since the simulator gets a random r giving rise to x (*i.e.*, $x = X(r)$) as an auxiliary input (whereas R is only given x). A sample test protocol is implicit in Okamoto's work [Oka] (where it is called a "post-test").

Theorem 5.9 (implicit in [Oka], explicit in [GV]) *There exists a public-coin sample test protocol. Furthermore, the number of messages exchanged in the protocol is linear in m.*

5.4 The Commitment Scheme

Now we use the above protocols to design problem-dependent commitments for all of **SZK**, and thereby prove Theorem 4.3. Let Π be a promise problem in **SZK**, let x be any string of length n, let $k = 2n$, and $\ell = n^{7c}$ for a sufficiently large constant c. Applying the reduction of Lemma 5.2, we obtain distributions (Z_0, Z_1) such that

- If $x \in \Pi_Y$, then $\Delta(Z_0, Z_1) \leq \ell \cdot 2^{-k} < 2^{-n}$.
- If $x \in \Pi_N$, then $\Delta(Z_0, Z_1) \geq 1 - 2^{-\ell}$.
- For all x, $\mathrm{H}(Z_0) = \mathrm{H}(Z_1)$ and both Z_0 and Z_1 are Δ-flat for $\Delta = \sqrt{\ell} \cdot \mathrm{poly}(n, k) < n^{4c}$ for a sufficiently large constant c.

Now we also define a new distribution Z as follows $Z(b, r) = Z_b(r)$. That is, Z outputs a random sample of Z_0 with probability $1/2$ and a random sample of Z_1 with probability $1/2$. Since $\mathrm{H}(Z_0) = \mathrm{H}(Z_1)$, we have $\mathrm{H}(Z_0) \leq \mathrm{H}(Z) \leq \mathrm{H}(Z_0) + 1$. We also claim that Z inherits the flatness of Z_0 and Z_1.

Claim 5.10 *Z is 3Δ-flat.*

Proof of claim: We need to show that a random sample $z \leftarrow Z$ is not $t \cdot 3\Delta$-typical for Z with probability at most 2^{-t^2}. For this, it suffices to separately bound the probabilities that z is not $t \cdot 3\Delta$-light and that z is not $t \cdot 3\Delta$-heavy. First we note that $t \cdot 3\Delta \geq 2t \cdot \Delta + 1$. For any z that is $(2t \cdot \Delta + 1)$-light for Z, we have

$$\Pr[Z_0 = z] \leq 2 \cdot \Pr[Z = z] \leq 2 \cdot 2^{-(2t\Delta+1)} \cdot 2^{-\mathrm{H}(Z)} \leq 2^{-2t\Delta} \cdot 2^{-\mathrm{H}(Z_0)}.$$

Similarly for Z_1. Therefore any such z is also $2t\Delta$-light for Z_0 and Z_1. Hence, if $z \leftarrow Z$, then z is $t \cdot (3\Delta)$-light with probability at most $2^{-(2t)^2}$.

Now we consider the heavy z's. Suppose that z is $(2t \cdot \Delta + 1)$-heavy for Z. Then

$$\max\{\Pr[Z_0 = z], \Pr[Z_1 = z]\} \geq \Pr[Z = z] \geq 2^{2t\Delta+1} \cdot 2^{-\mathrm{H}(Z)} \geq 2^{2t\Delta+1} \cdot 2^{-(\mathrm{H}(Z_0)+1)}.$$

Thus any such z is $2t \cdot \Delta$-heavy for either Z_0 or Z_1, wlog say Z_0. The probability that Z_0 outputs a string that $2t \cdot \Delta$-heavy for Z_0 is at most $2^{-(2t)^2}$, by Δ-flatness. However we also need to bound the probability that Z_1 outputs such a string. Let H_0 be the set of strings that are $2t \cdot \Delta$-heavy for Z_0. These strings have total probability mass at least $|H_0| \cdot 2^{-\mathrm{H}(Z_0)+2t\Delta}$ under Z_0, and probability mass at most $2^{-(2t)^2}$ by Δ-flatness. Thus, $|H_0| \leq 2^{-(2t)^2} \cdot 2^{\mathrm{H}(Z_0)-2t\Delta}$. Then

$$\Pr[Z_1 \in H_0] \leq \Pr[Z_1 \text{ is } 2t\Delta\text{-heavy}] + |H_0| \cdot 2^{-\mathrm{H}(Z_1)+2t\Delta} \leq 2^{-(2t)^2} + 2^{-(2t)^2}.$$

We can do an identical analysis for the strings H_1 that are $2t\Delta$-heavy for Z_1. Then

$$\begin{aligned}
\Pr[Z \in H_0 \cup H_1] &= \frac{1}{2}\left(\Pr[Z_0 \in H_0] + \Pr[Z_1 \in H_0] + \Pr[Z_0 \in H_1] + \Pr[Z_1 \in H_1]\right) \\
&\leq \frac{1}{2}\left(2^{-(2t)^2} + 2 \cdot 2^{-(2t)^2} + 2 \cdot 2^{-(2t)^2} + 2^{-(2t)^2}\right) \\
&= 3 \cdot 2^{-(2t)^2}
\end{aligned}$$

In total, we see that the probability that Z is not $t \cdot (3\Delta)$-typical is at most $2^{-(2t)^2} + 3 \cdot 2^{-(2t)^2} \leq 2^{-t^2}$, for $t \geq 1$. □

We also set $t = n$, and define the problem-dependent commitment scheme (S, R) as follows:

Commit Phase $(S_1(b), R_1)(x)$**:** 1. S_1 and R_1 execute the sample generation protocol on input $(Z, 3\Delta, t)$ to obtain output z.

2. S_1 chooses (c, r) uniformly s.t. $Z(c, r) = z$, and sends $d = b \oplus c$ to R.
3. The commitment is defined as the pair (z, d).

Intuitively, if Z_0 and Z_1 are statistically close, then a random sample z of Z is nearly equally likely to have come from Z_0 or Z_1, so the bit c is random and hides b.

Valid Commitments The promise problem of valid commitments is defined to be $\textsc{Val} = (\textsc{Val}_Y, \textsc{Val}_N)$ where

$$\begin{aligned} \textsc{Val}_Y &= \{(x, (d, z), b) : z \text{ is } t\Delta\text{-typical for } Z_{d\oplus b}\} \\ \textsc{Val}_N &= \{(x, (d, z), b) : z \text{ is } 6\sqrt{t\Delta} \cdot \Delta\text{-light for } Z_{d\oplus b}\} \end{aligned}$$

Reveal Phase $(S_2, R_2)(x, (d, z), b)$**:** 1. S_2 and R_2 execute the sample test protocol on input $(Z_{d\oplus b}, z, \Delta, t)$, and R_2 accepts or rejects according to its outcome.

Now we prove that the protocol has the desired properties.

1. (Efficiency) Follows from efficiency of the sample generation and sample test protocols.

2. (Completeness) By the completeness of the sample generation protocol, the string z generated in the $(S_1(b), R_1)(x)$ has statistical difference at most $m \cdot 2^{-t^2} < 2^{-n}$ from Z. Thus (c, r) has statistical difference at most 2^{-n} from uniform. If (c, r) were uniformly distributed, then by the Δ-flatness of $Z_{c\oplus d}$, the probability (over r) that $z = Z_c(r)$ is $t\Delta$-typical for $Z_c = Z_{d\oplus b}$ is least $1 - 2^{-t^2} > 1 - 2^{-n}$. Therefore, $(x, (d, z), b) \in \textsc{Val}_Y$ with probability at least $1 - 2 \cdot 2^{-n}$.

3. (Validity Tests) The completeness and soundness of the sample test protocol show that (S_2, R_2) is an interactive proof system for $\textsc{Val}$. To see that $\textsc{Val}$ is in **AM**, note that proving that an instance $(x, (d, z), b) \in \textsc{Val}_Y \cup \textsc{Val}_N$ is a YES instance amounts to an approximate lower bound on the size of the set $\{r : Z_{d\oplus b}(r) = z\}$ (since $6\sqrt{t\Delta} \cdot \Delta > t\Delta + 1$).

4. (Zero Knowledge) The zero-knowledge condition follows from the zero-knowledge conditions of the sample generation and sample test protocols. Specifically, the simulator $M(x, b)$ chooses a uniformly random (c, r), sets $z = Z(c, r)$ and $d = b \oplus c$, runs the simulator for the sample generation protocol on input $(Z, 3\Delta, t, z)$ to obtain a transcript γ_1, and runs the simulator for the sample test protocol on (Z_c, z, Δ, t, r) to obtain a transcript γ_2, and outputs (γ_1, d, γ_2).

5. (Statistically hiding on YES instances) The only dependence of R_1's view on the bit b is in the value $d = b \oplus c$, where c is selected based on z, according to the conditional distribution of σ given that $Z_\sigma = z$, where σ is a random bit. In the case of a YES instance, Z_0 and Z_1 have statistical difference at most 2^{-n}, and we have seen above that z has statistical difference at most 2^{-n} from a random sample of Z. Together these imply the pair (z, c) has statistical difference at most $2 \cdot 2^{-n}$ from (z, σ) where σ is a random bit independent of z, and thus R_1's view in case $b = 0$ is indistinguishable from R_1's view in case $b = 1$.

6. (Statistically binding on NO instances) Let $T = \{z : z \text{ is not } 6\sqrt{t\Delta} \cdot \Delta\text{-light for } Z_0 \text{ nor for } Z_1\}$. By Lemma 5.3,

$$|T| \leq \frac{2^{\mathrm{H}(Z_0)}}{2^{\ell - 6\sqrt{t\Delta}\cdot\Delta}} \leq 2^{-6\sqrt{t\Delta}\cdot\Delta} \cdot 2^{\mathrm{H}(Z)},$$

where the last inequality is because $\ell = n^{7c} > 12n^{6c+.5} > 12\sqrt{t\Delta}\cdot\Delta$. By the second soundness condition of the sample generation protocol, the probability that the output z is in T is at most $2^{-\Omega(t^2)} < 2^{-n}$. If the output is not in T, then for any d, there is at most one value of b such that z is not $6\sqrt{t\Delta} \cdot \Delta$-light for $Z_{d\oplus b}$. That is, there is at most one value of b such that $(x, (z, d), b) \notin \textsc{Val}_N$, as desired.

■

6 Putting it Together

Now we can put together the results proven in the previous three sections and establish Theorems 1.2, 3.3, 3.5, 4.2

Theorem 6.1 *For a promise problem* Π*, the following conditions are equivalent:*

1. $\Pi \in$ **HVZK**.
2. $\Pi \in$ **IP** *and* Π *satisfies the* Conditional Pseudoentropy Characterization.
3. $\Pi \in$ **IP** *and* Π *satisfies the* SZK/OWF Characterization.
4. $\Pi \in$ **IP** *and* Π *satisfies the* Indistinguishability Characterization.
5. $\Pi \in$ **IP** *and* Π *has a public-coin computationally hiding problem-dependent commitment scheme in the sense of Definition 4.1. Moreover the sender can be implemented in probabilistic polynomial time given an* **NP** *oracle.*
6. Π *is in* **ZK**.
7. Π *has a public-coin computational zero-knowledge proof with a black-box simulator and perfect completeness.*
8. Π *has a public-coin computational zero-knowledge proof with a black-box simulator, where on any input* x*, the prover strategy* P_x *can be computed in probabilistic polynomial time given an* **NP** *oracle and an oracle for* $\hat{P}_x$*, where* $\hat{P}$ *is the prover in any interactive proof system for* Π*. In particular, if* $\Pi \in$ **NP** *(or even* $\Pi \in$ **AM***), then* P_x *can be computed in probabilistic polynomial time with an* **NP** *oracle.*

Proof:

1 $\Rightarrow$ **2** This is Lemma 3.6.

2 $\Rightarrow$ **3** This is Lemma 3.9.

3 $\Rightarrow$ **5** This is Lemma 4.4.

5 ⇒ 7 Suppose $\Pi \in \mathbf{IP}$ and Π has a problem-dependent commitment scheme. By Lemma 4.8, Π has a public-coin honest-verifier zero-knowledge proof. We can convert this into a public-coin proof system with perfect completeness using the transformation of Fürer et al. [FGM$^+$], which preserves honest-verifier zero knowledge. Finally, by Theorem 4.9, this can be converted into a public-coin (cheating-verifier) zero-knowledge proof with a black-box simulator and perfect completeness.

5 ⇒ 8 This is proven the same way as in the previous item, except we omit the transformation of Fürer et al. [FGM$^+$] (which seems to increase the prover complexity too much for our purposes). For bounding the prover complexity, we first note that if $\Pi \in \mathbf{IP}$, then a (variant of) the Goldwasser–Sipser [GS] transformation converts any interactive proof $(\hat{P}, \hat{V})$ for Π into a public-coin interactive proof where the prover on input x can be implemented in probabilistic polynomial time given an **NP** oracle and oracle access to $\hat{P}_x$. Then Lemma 4.8 preserves this prover complexity because the sender in the problem-dependent commitment can be implemented in probabilistic polynomial time with an **NP** oracle, as does Theorem 4.9.

7/8 ⇒ 6 ⇒ 1 These are immediate from the definitions.

2 ⇔ 4 This is Lemmas 3.12 and 3.13.

■

7 Applications and Extensions

7.1 The Ostrovsky–Wigderson Theorems

As described in the Introduction, the approach of this paper and in particular the SZK/OWF Characterization, are inspired by the work of Ostrovsky and Wigderson [OW], who showed that "nontriviality" of **ZK** implies "some form of one-way functions". In this section, we show how our results can be used to give new, more modular proofs of the Ostrovsky–Wigderson theorems.

The two Ostrovsky–Wigderson theorems are obtained by two different interpretations of "nontriviality" and "some form of one-way functions." In their first theorem (mentioned in the Introduction), both are interpreted in a weak sense:

Theorem 7.1 ([OW, Thm 1]) *If* $\mathbf{HVZK} \neq \mathbf{BPP}$*, then there exists a poly-time auxiliary-input family of functions* $\{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ *that is not "easy to invert". That is, for every PPT* A *and polynomial* $r(n)$*, there exists an infinite set* $I \subseteq \{0,1\}^*$ *such that*

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

for all $x \in I$*.*

We point out that the theorem above refers to *uniform* PPT inverters A; to obtain functions that are not easy to invert by nonuniform algorithms, the hypothesis should be replaced by $\mathbf{HVZK} \not\subset \mathbf{P/poly}$.

In their second theorem, both conditions are interpreted in a strong sense:

Theorem 7.2 ([OW, Thm 2], informally stated) *If* **HVZK** *contains a "hard-on-average" problem, then (standard) one-way functions exist.*

Here we focus on the first theorem, and thus omit a formal definition of "hard-on-average" problem.

We begin by observing that the SZK/OWF CHARACTERIZATION immediately implies a stronger form of one-way functions than given by Theorem 7.1 under the stronger hypothesis that **HVZK** $\neq$ **HVSZK**.

Theorem 7.3 *If* **HVZK** $\neq$ **HVSZK**, *then there exists an auxiliary-input one-way function on some infinite set I. That is, there is a poly-time auxiliary-input family of functions $\{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ and an infinite set I such that for every nonuniform PPT A and every polynomial $r(n)$, we have*

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

for all sufficiently long $x \in I$.

The key difference between the conclusions of Theorem 7.1 and Theorem 7.3 is that the order of quantifiers between the adversary A and the infinite set I is reversed. In the former, the infinite set indices x for which the adversary fails to invert f_x can depend on the adversary A, whereas in the latter, there is a fixed infinite set of indices such that f_x is hard for *all* polynomial-time adversaries A.

Proof of Theorem 7.3: Suppose **HVZK** $\neq$ **HVSZK**, and let Π be any promise problem in **HVZK** $\setminus$ **HVSZK**. By Theorem 6.1, Π satisfies the SZK/OWF CHARACTERIZATION. That is, there is a set I such that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in **SZK** and there exists an auxiliary-input one-way function on I. We claim that I is infinite (which suffices to complete the proof). Suppose for sake of contradiction that I is finite. Since $\Pi' \in$ **SZK** and Π and Π' differ on only a finite set of inputs, we conclude that $\Pi \in$ **SZK** $\subseteq$ **HVSZK**. (The statistical zero-knowledge proof for Π is the same as the statistical zero-knowledge proof for Π', except we hardwire the set I into the verifier and simulator, have the verifier immediately accept inputs $x \in I$, and have the prover send nothing on such inputs.) This contradicts the choice of Π. ■

We now give an alternate proof of Theorem 7.1.

Proof of Theorem 7.1: Suppose that **HVZK** $\neq$ **BPP**. Then either **HVZK** $\neq$ **HVSZK** or **HVSZK** $\neq$ **BPP**. In the first case, we are done by Theorem 7.3. Thus, we need only show that **HVSZK** $\neq$ **BPP** implies the existence of an auxiliary-input family of functions that is not easy to invert. This can be done using the techniques of Ostrovksy [Ost], who proved an **HVSZK**-analogue of Theorem 7.2. (Ostrovsky's paper was a precursor to the Ostrovsky–Wigderson result, and has a much simpler proof.) Below we give an alternative proof, utilizing the complete problems for **SZK**.

Suppose that **HVSZK** $\neq$ **BPP**. In [GSV2], it was shown that this implies that for some constant $\delta > 0$, the following promise problem IMAGE DENSITY (ID) is not in **BPP**:[10]

$$\begin{aligned} \mathrm{ID}_Y &= \{C : \Delta(C(U_m), U_n) \leq 2^{-|C|^\delta}\} \\ \mathrm{ID}_N &= \{C : \mathrm{Image}(C) \leq 2^{-|C|^\delta} \cdot 2^n\}, \end{aligned}$$

[10]Specifically, [GSV2, Lemma 5.1] shows that the **HVSZK**-complete problem ENTROPY DIFFERENCE Cook-reduces to a promise problem called ENTROPY APPROXIMATION, and [GSV2, Lemma 3.2] shows that ENTROPY APPROXIMATION (Karp-)reduces to IMAGE DENSITY. (We remark that both ENTROPY APPROXIMATION and IMAGE DENSITY are complete for **NISZK**, the class of problems having noninteractive statistical zero-knowledge proofs [DDPY, GSV2].)

where $C : \{0,1\}^m \to \{0,1\}^n$ is a circuit, and $\text{Image}(C) \subseteq \{0,1\}^n$ is its image. Thinking of C as a sampling circuit for the distribution $C(U_m)$, we see that IMAGE DENSITY is a restricted version of STATISTICAL DIFFERENCE, where one of the distributions is fixed to be U_n, and in NO instances the distribution is not only far from uniform but also has small support. However, in what follows it is more useful to think of the circuit C as specifying a function rather than a distribution.

We define an auxiliary-input family of functions $\{f_C\}$, where the function indexed by circuit $C : \{0,1\}^m \to \{0,1\}^n$ is simply $f_C = C$. Suppose, for sake of contradiction, that this family of functions is easy to invert. That is, there exists a PPT A and a polynomial r such that for every circuit $C : \{0,1\}^m \to \{0,1\}^n$, $\Pr\left[A(C, C(U_m)) \in C^{-1}(C(U_m))\right] \geq 1/r(|C|)$. Then we obtain the following decision procedure for ID: On input C, where $C : \{0,1\}^m \to \{0,1\}^n$ is a circuit, we simply choose $x \leftarrow U_n$ and check if $C(A(C,x)) = x$. That is, we see if A successfully finds a preimage of a uniformly chosen element of $\{0,1\}^n$.

If $C \in \text{ID}_Y$, then U_n has exponentially small statistical difference from $C(U_m)$. Since A successfully inverts C on $C(U_m)$ with probability at least $1/r(|C|)$, it will successfully invert C on U_n with probability at least $1/r(|C|) - 1/2^{|C|^\delta} \geq 1/(2r(|C|))$. On the other hand, if $C \in \text{ID}_N$, then the image of C has exponentially small density and hence the probability that U_n has any preimage under C is at most $2^{-|C|^\delta}$. Consequently, A will successfully invert C on U_n with exponentially small probability. By standard amplfication, we conclude that $\text{ID} \in \mathbf{BPP}$. ■

The above proof also illustrates why one only obtains a family of functions that is not easy to invert, rather than the stronger notion of auxiliary-input one-way functions achieved in Theorem 7.3. The reason is that the supposed inverter for the family of functions is used to construct a **BPP** algorithm for IMAGE DENSITY and thereby for all of **SZK**. The hypothesis that $\mathbf{SZK} \neq \mathbf{BPP}$ only seems to guarantee that for every inverter A there exists an infinite set I_A of instances on which this procedure fails, not that there exists a fixed infinite set I of "hard" instances on which the procedure fails for any A. For example, an inverter A running in time n^2 may be able to succeed on a larger set of instances than an inverter running in time n, and one running in time n^3 may succeed on an even larger set of instances, and so on. Ultimately, the set of instances which are hard for *all* polynomial-time A may be empty. Intuitively, we avoided this difficulty in the proof of Lemma 3.9 by using an *information-theoretic* condition (namely whether $\text{H}(X|Y) \geq r$) to separate the **SZK** instances from the OWF instances.

As for an alternative proof of Theorem 7.2, it was already shown in [SV, Thm 5.12] that if a hard-on-average problem satisfies the INDISTINGUISHABILITY CHARACTERIZATION, then one-way functions exist. (The intuition is as follows: according to the INDISTINGUISHABILITY CHARACTERIZATION, YES instances give rise to samplable distributions X and Y that computationally indistinguishable, and NO instances give rise to X and Y that are statistically far. However, the fact that a problem is hard-on-average means that YES instances and NO instances are computationally indistinguishable from each other, under some samplable distribution on instances. Thus, we can obtain two samplable distributions that are both computationally indistinguishable and statistically far apart, which implies the existence of one-way functions by [Gol1].) Combining this with our result that every problem in **HVZK** satisfies the INDISTINGUISHABILITY CHARACTERIZATION (Thm. 6.1), we obtain Theorem 7.2.

We note that these new proofs of Theorems 7.1 and 7.2 only use our results from Section 3, namely that every problem in **HVZK** satisfies the SZK/OWF CHARACTERIZATION and the INDISTINGUISHABILITY CHARACTERIZATION. Our results in the converse direction, from Section 4, are not needed.

7.2 Monotone Closure

In this section, we use our results to prove closure properties of **ZK**. We begin by noting that the fact that **ZK** is closed under intersection is immediate: to prove that $x \in \Pi_Y \cap \Gamma_Y$ for promise problems $\Pi, \Gamma \in \mathbf{ZK}$, the prover can prove that $x \in \Pi_Y$ using the zero-knowledge proof for Π and then prove that $x \in \Gamma_Y$ using the zero-knowledge proof for Γ, and the verifier accepts only if both proofs are convincing. The analogous approach for union, however, does not work. In particular, proving that $x \in \Pi_Y \cup \Gamma_Y$ seems to require the prover to reveal whether $x \in \Pi_Y$ or $x \in \Gamma_Y$, and thus the proof system may not be zero knowledge.

In this section, we show **ZK** is indeed closed under union. More generally, for every $\Pi \in \mathbf{ZK}$, we give zero-knowledge proofs for arbitrary monotone boolean formulae over statements about membership in Π, where the formula can even be specified as part of the common input. Our constructions are direct generalizations of the techniques of [DDPY, SV], who proved analogous closure properties for **SZK** and subclasses of **SZK**. (In fact, since **SZK** is closed under complement [Oka], its closure properties extend even to non-monotone formulae.) Indeed, we simply replace Statistical Difference in the construction of [SV] with the Indistinguishability Characterization.

We begin with closure under union, only sketching the proof since it is subsumed by the more general result presented later.

Theorem 7.4 **ZK** *is closed under union.*

Proof Sketch: By Theorem 3.3, a promise problem is in **ZK** if and only if it is in **IP** and it satisfies the Indistinguishability Characterization. We know that **IP** is closed under union, so it suffices to show that the class of problems satisfying the Indistinguishability Characterizationis closed under union.

Suppose Π and Γ satisfy the Indistinguishability Characterization. From any instance w, we obtain two samplable distributions X_0, X_1 that are computationally indistinguishable if $w \in \Pi_Y$ and statistically far if $w \in \Pi_N$. Similarly, we obtain Y_0, Y_1 for Γ. Consider the following two distributions:

Z_0**:** Choose $b, c \leftarrow \{0,1\}$ such that $b \oplus c = 0$. Sample $x \leftarrow X_b$ and $y \leftarrow Y_c$. Output (x, y).

Z_1**:** Choose $b, c \leftarrow \{0,1\}$ such that $b \oplus c = 1$. Sample $x \leftarrow X_b$ and $y \leftarrow Y_c$. Output (x, y).

Suppose $w \in (\Pi \cup \Gamma)_Y = \Pi_Y \cup \Gamma_Y$; wlog say $w \in \Pi_Y$. Then X_0 and X_1 are computationally indistinguishable. Since the only difference between Z_0 and Z_1 is that X_0 and X_1 are swapped (that is, if we replace X_b with $X_{\neg b}$ in the definition of Z_0, we obtain Z_1), it follows that Z_0 and Z_1 are computationally indistinguishable.

Suppose that $w \in (\Pi \cup \Gamma)_N = \Pi_N \cap \Gamma_N$. It is shown in [SV, Prop. 3.6] that $\Delta(Z_0, Z_1) = \Delta(X_0, X_1) \cdot \Delta(Y_0, Y_1)$, so $\Delta(Z_0, Z_1) \geq (2/3) \cdot (2/3) = 4/9$. This suffices to prove that $\Pi \cup \Gamma$ satisfies the Indistinguishability Characterization. □

We now present some definitions (closely following [SV]) to formalize the more general monotone closure properties we will obtain. Specifically, in order to deal with instances of promise problems that violate the promise, we will work with an extension of boolean algebra that includes an additional"ambiguous" value $\star$.

Definition 7.5 *A* partial assignment *to variables $v_1, \ldots, v_k$ is a k-tuple $\overline{a} = (a_1, \ldots, a_k) \in \{0, 1, \star\}^k$. For a propositional formula (or circuit) ϕ on variables $v_1, \ldots, v_k$, the evaluation $\phi(\overline{a})$ is recursively defined as follows:*

$$v_i(\overline{a}) = a_i \qquad (\phi \wedge \psi)(\overline{a}) = \begin{cases} 1 & \text{if } \phi(\overline{a}) = 1 \text{ and } \psi(\overline{a}) = 1 \\ 0 & \text{if } \phi(\overline{a}) = 0 \text{ or } \psi(\overline{a}) = 0 \\ \star & \text{otherwise} \end{cases}$$

$$(\neg\phi)(\overline{a}) = \begin{cases} 1 & \text{if } \phi(\overline{a}) = 0 \\ 0 & \text{if } \phi(\overline{a}) = 1 \\ \star & \text{if } \phi(\overline{a}) = \star \end{cases} \qquad (\phi \vee \psi)(\overline{a}) = \begin{cases} 1 & \text{if } \phi(\overline{a}) = 1 \text{ or } \psi(\overline{a}) = 1 \\ 0 & \text{if } \phi(\overline{a}) = 0 \text{ and } \psi(\overline{a}) = 0 \\ \star & \text{otherwise} \end{cases}$$

Note that $\phi(\overline{a})$ equals 1 (resp., 0) for some partial assignment $\overline{a}$, then $\phi(\overline{a}')$ also equals 1 (resp., 0) for every boolean $\overline{a}'$ obtained by replacing every $\star$ in $\overline{a}$ with either a 0 or 1. The converse, however, is not true: The formula $\phi = v \vee \neg v$ evaluates to 1 on every boolean assignment, yet is not 1 when evaluated at $\star$. Thus, the "law of excluded middle" $\phi \vee \neg\phi \equiv 1$ no longer holds in this setting. However, other identities in boolean algebra such as De Morgan's laws (e.g. $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$) do remain true.

Definition 7.6 *For a promise problem Π, the* characteristic function *of Π is the map $\chi_\Pi : \{0,1\}^* \to \{0, 1, \star\}$ given by*

$$\chi_\Pi(x) = \begin{cases} 1 & \text{if } x \in \Pi_Y \\ 0 & \text{if } x \in \Pi_N \\ \star & \text{otherwise} \end{cases}$$

Definition 7.7 *For any promise problem Π and constant $\delta > 0$, we define a new promise problem $\mathrm{Mon}_\delta(\Pi)$ as follows:*

$$\begin{aligned} \mathrm{Mon}_\delta(\Pi)_Y &= \{(\phi, x_1, \ldots, x_k) : \phi(\chi_\Pi(x_1), \ldots, \chi_\Pi(x_k)) = 1 \text{ and } \forall i \ |x_i| \geq n^\delta\} \\ \mathrm{Mon}_\delta(\Pi)_N &= \{(\phi, x_1, \ldots, x_k) : \phi(\chi_\Pi(x_1), \ldots, \chi_\Pi(x_k)) = 0 \text{ and } \forall i \ |x_i| \geq n^\delta\}. \end{aligned}$$

where ϕ is a monotone *k-ary propositional formula, and $n = |(\phi, x_1, \ldots, x_k)|$.*

The condition $|x_i| \geq n^\delta$ is a technicality due to the fact that the security of zero-knowledge proofs is defined with respect to the input length. Intuitively, we will be constructing zero-knowledge proofs for instances of $\mathrm{Mon}_\delta(\Pi)$ of length $n = |(\phi, x_1, \ldots, x_k)|$, but these will be built by using zero-knowledge proofs (or the resulting Indistinguishability Characterization) for the individual x_i's. Hence to achieve security in terms of n, we will need the x_i's to be of length polynomially related to n. Naturally, this entire issue disappears if one works with a security-parametrized definition of zero knowledge. (See Remark 5 at the end of Section 2.5.)

Theorem 7.8 *For any promise problem $\Pi \in$ **SZK** and any $\delta > 0$, $\mathrm{Mon}_\delta(\Pi) \in$ **SZK**.*

Proof: First we note that **IP** is closed under $\mathrm{Mon}_\delta(\cdot)$: to prove that $(\phi, x_1, \ldots, x_k) \in \mathrm{Mon}_\delta(\Pi)_Y$, by monotonicity of ϕ, it suffices to prove that a subset of the x_i's are in Π_Y. Thus, by Theorem 6.1 we need only show that if Π satisfies the Indistinguishability Characterization, then $\mathrm{Mon}_\delta(\Pi)$ satisfies the Indistinguishability Characterization.

Suppose we are given an instance $(\phi, x_1, \ldots, x_k)$ of $\mathrm{Mon}_\delta(\Pi)$. Since Π satisfies the Indistinguishability Characterization, from each each x_i we can efficiently construct two samplable

```
Sample(ψ, b)
    If ψ = v_i, sample z ← X_b^i.
    If ψ = τ ∧ μ,
        Sample z_1 ← Sample(τ, b);
        Sample z_2 ← Sample(μ, b);
        Let z = (z_1, z_2).
    If ψ = τ ∨ μ,
        Choose c, d ← {0,1} subject to c ⊕ d = b;
        Sample z_1 ← Sample(τ, c);
        Sample z_2 ← Sample(μ, d);
        Let z = (z_1, z_2).
    Output z.
```

Figure 1:

distributions X_0^i, X_1^i that are either computationally indistinguishable or statistically far apart, depending on whether x_i is a YES or NO instance of Π. The idea is to recursively combine these distributions according the formula ϕ, to eventually obtain two samplable distributions that are either computationally indistinguishable or statistically far apart according to whether $(\phi, x_1, \ldots, x_k)$ is a YES or NO instance of $\mathrm{Mon}_\delta(\Pi)$. At OR gates, we will combine distributions via the construction used to prove closure under union above. At AND gates, we will combine distributions via the Direct Product construction $(X_0, X_1), (Y_0, Y_1) \mapsto (X_0 \otimes X_1, Y_0 \otimes Y_1)$; this has the property that the resulting pair of distributions are computationally indistinguishable if either of the initial pairs is indistinguishable, and are statistically far apart if both initial pairs are statistically far apart. This is precisely the construction used in [SV, Lemma 4.10], except that we need to work with computational indistinguishability rather than statistical closeness and the treatment of AND and OR is swapped because our Indistinguishability Characterization corresponds to the complement of the definition of Statistical Difference used there.

Now we make this construction and its analysis precise. First, by taking direct products, we may assume that the distributions $\{X_0^i, X_1^i\}$ satisfy:

- $x_i \in \Pi_Y \Rightarrow X_0^i$ and X_1^i are computationally indistinguishable. More precisely, no circuit of size s can distinguish between X_0^i and X_1^i with advantage greater than ε for some $s = (n^\delta)^{\omega(1)} = n^{\omega(1)}$ and $\varepsilon = 1/(n^\delta)^{\omega(1)} = 1/n^{\omega(1)}$.
- $x_i \in \Pi_N \Rightarrow \Delta(X_0^i, X_1^i) \geq 1 - 2^{-n}$.

Consider the randomized recursive procedure Sample(ψ, b) in Figure 1 which takes a subformula ψ of $\phi = \phi(v_1, \ldots, v_n)$ and a bit $b \in \{0,1\}$ as input. Executing Sample(ϕ, b) for $b \in \{0,1\}$ takes time polynomial in n, because the total number of recursive calls is equal to the number of subformulae of ϕ.

Let $\overline{a} = (\chi_\Pi(x_1), \ldots, \chi_\Pi(x_k))$. From [SV, Claim 4.11], it follows that if $\phi(\overline{a}) = 0$, then $\Delta(\text{Sample}(\phi, 0), \text{Sample}(\phi, 1)) \geq 1 - |\phi| \cdot 2^{-n} \geq 2/3$.

Thus, to establish the Indistinguishability Characterization, we need only show that if $\phi(\overline{a}) = 1$, then Sample$(\phi, 0)$ and Sample$(\phi, 1)$ are computationally indistinguishable.

Claim 7.9 *For every subformula ψ such that $\psi(\overline{a}) = 1$, no circuit of size s can distinguish between* $\text{Sample}(\psi, 0)$ *and* $\text{Sample}(\psi, 1)$ *with advantage greater than $|\psi| \cdot \varepsilon$.*

Proof of claim: The proof is by induction on the subformulas of ϕ.

Base Case: $\psi = v_i$ The condition $\psi(\overline{a}) = 1$ means that $\chi_\Pi(x_i) = 1$, and hence no circuit of size s can distinguish between $\text{Sample}(\psi, 0) = X_0^i$ and $\text{Sample}(\psi, 1) = X_1^i$ with advantage greater than ε.

Inductive Case I: $\psi = \tau \wedge \mu$ The condition $\psi(\overline{a}) = 1$ means that $\tau(\overline{a}) = 1$ and $\mu(\overline{a}) = 1$. We have $\text{Sample}(\psi, b) = \text{Sample}(\tau, b) \otimes \text{Sample}(\mu, b)$ for $b = 0, 1$. Consider the hybrid distribution $H = \text{Sample}(\tau, 0) \otimes \text{Sample}(\mu, 1)$. Suppose for sake of contradiction that there exists a circuit D of size s that distinguishes between $\text{Sample}(\psi, 0)$ and $\text{Sample}(\psi, 1)$ with advantage greater than $|\psi| \cdot \varepsilon \geq (|\tau| + |\mu|) \cdot \varepsilon$. Then D must distinguish between $\text{Sample}(\psi, 0)$ and H with advantage greater than $|\tau| \cdot \varepsilon$ or distinguish between H and $\text{Sample}(\psi, 1)$ with advantage greater than $|\mu| \cdot \varepsilon$. By fixing one of the inputs to D, we obtain a circuit of size s that distinguishes between $\text{Sample}(\tau, 0)$ and $\text{Sample}(\tau, 1)$ with advantage greater than $|\tau| \cdot \varepsilon$ or between $\text{Sample}(\mu, 0)$ and $\text{Sample}(\mu, 1)$ with advantage greater than $|\mu| \cdot \varepsilon$. This contradicts the inductive hypothesis.

Inductive Case II: $\psi = \tau \vee \mu$ The condition $\psi(\overline{a}) = 1$ means that $\tau(\overline{a}) = 1$ or $\mu(\overline{a}) = 1$. Without loss of generality, say that $\tau(\overline{a}) = 1$. $\text{Sample}(\psi, b)$ is equivalent to the distribution that w.p. $1/2$ outputs a sample of $\text{Sample}(\tau, b) \otimes \text{Sample}(\mu, 0)$ and w.p. $1/2$ outputs a sample of $\text{Sample}(\tau, \neg b) \otimes \text{Sample}(\mu, 1)$. Thus if a circuit D of size s distinguishes between $\text{Sample}(\psi, 0)$ and $\text{Sample}(\psi, 1)$ with advantage $|\psi| \cdot \varepsilon$, then by averaging, D must also distinguish between $\text{Sample}(\tau, 0) \otimes \text{Sample}(\mu, 0)$ and $\text{Sample}(\tau, 1) \otimes \text{Sample}(\mu, 0)$, or between $\text{Sample}(\tau, \neg 0) \otimes \text{Sample}(\mu, 1)$ and $\text{Sample}(\tau, \neg 1) \otimes \text{Sample}(\mu, 1)$, with advantage at least $|\psi| \cdot \varepsilon$. By fixing the second input to D, we obtain a size s distinguisher between $\text{Sample}(\tau, 0)$ and $\text{Sample}(\tau, 1)$ with advantage at least $|\psi| \cdot \varepsilon > |\tau| \cdot \varepsilon$, contradicting the inductive hypothesis.

□

■

Theorem 7.8 can be also viewed as demonstrating that **ZK** is closed under a type of polynomial-time reducibility, which is formalized by the following two definitions.

Definition 7.10 (truth-table reduction [LLS]): *We say a promise problem* Π truth-table reduces *to a promise problem* Γ *if there exists a (deterministic) polynomial-time computable function f, which on input x produces a tuple $(y_1, \ldots, y_k)$ and a boolean circuit C (with k input gates) such that*

$$\begin{aligned} x \in \Pi_Y &\Rightarrow C(\chi_\Gamma(y_1), \ldots, \chi_\Gamma(y_k)) = 1 \\ x \in \Pi_N &\Rightarrow C(\chi_\Gamma(y_1), \ldots, \chi_\Gamma(y_k)) = 0 \end{aligned}$$

We call such a reduction non-shrinking *if we have $\forall i \ |y_i| \geq |x|^\delta$, for some constant $\delta > 0$.*

In other words, a truth-table reduction for promise problems is a nonadaptive Cook reduction which is allowed to make queries that violate the promise, but still must have an unambiguous

output (in the strong sense formalized by Definition 7.5). We further consider the case where we restrict the complexity of computing the output of the reduction from the queries:

Definition 7.11 ($\mathbf{NC}^1$ truth-table reductions): *A truth-table reduction f between promise problems is an* $\mathbf{NC}^1$ truth-table reduction *if the circuit C produced by the reduction on input x has depth bounded by $c_f \log |x|$, where c_f is a constant independent of x. It is* monotone *if the circuit C has only AND and OR gates (but no negations).*

With these definitions, we can restate Theorem 7.8 as follows:

Corollary 7.12 **ZK** *is closed under non-shrinking, monotone* $\mathbf{NC}^1$ *truth-table reductions.*

Proof: Any circuit of size s and depth d can be efficiently "unrolled" into a formula of size $2^d \cdot s$. Hence, a non-shrinking $\mathbf{NC}^1$ truth-table reduction from Γ to Π (with parameter δ) gives rise to a non-shrinking Karp reduction from Γ to $\mathrm{Mon}_{\delta/c}(\Pi)$ (where the reduction produces outputs of length at most n^c). Since **ZK** is closed under $\mathrm{Mon}(\cdot)$ and non-shrinking Karp reductions, it is also closed under $\mathbf{NC}^1$ truth-table reductions. ■

As shown in [SV], closure under such reductions has a consequence for *knowledge complexity* [GMR, GP], which is a framework for quantifying the amount $k(n)$ of knowledge leaked in an interactive proof system. Zero knowledge is the special case where $k(n) = 0$. There are various formalizations of the notion of knowledge complexity, most of which measure the number of bits of "help" that a simulator needs to simulate the verifier's view of the interaction. The simplest (but not entirely satisfactory) formulation is the following:

Definition 7.13 (knowledge complexity in the hint sense) *An interactive proof system (P, V) for a promise problem Π is said to have (honest-verifier)* knowledge complexity $k(n)$ in the hint sense $\kappa : \mathbb{N} \to \mathbb{N}$ *if there is a function $h : \Pi_Y \to \{0,1\}^*$, and a probabilistic polynomial-time algorithm S, such that for all $x \in \Pi_Y$*

1. $|h(x)| = \kappa(|x|)$.
2. $\langle P, V\rangle(x)$ *and* $S(x, h(x))$ *are computationally indistinguishable.*

$\mathbf{KC_{hint}}(k(n))$ *denotes the class of problems having interactive proofs with knowledge complexity $k(n)$ in the hint sense.*

We have restricted the definition to honest verifiers for simplicity, but the definition and our results can be extended to the cheating-verifier one as well. Using Corollary 7.12, we can prove a collapse in this hierarchy:

Theorem 7.14 *For every polynomially bounded function $k(n)$, $\mathbf{KC_{hint}}(k(n)+\log n) = \mathbf{KC_{hint}}(k(n))$. In particular, $\mathbf{KC_{hint}}(O(\log n)) = \mathbf{ZK}$.*

Proof Sketch: The proof is identical to the analogous result for **SZK** in [SV, Thm 4.15]. That proof uses the fact that **HVSZK** is closed under $\mathbf{NC}^1$ truth-table reductions. By inspection, the reduction used in the proof is non-shrinking and monotone (in fact the circuit produced simply computes the OR of its inputs). □

In addition, as shown in [Vad1, Sec. 4.6.2, Cor. 6.5.2] for **SZK**, many of our other results about **ZK** extend to $\mathbf{KC_{hint}}$. In particular, we obtain an equivalence between the honest-verifier and cheating-verifier definitions of $\mathbf{KC_{hint}}$, between private coins and public coins, etc. We omit the formal statements here.

7.3 Expected Polynomial-Time Simulators and weak-ZK

Recall that, following Goldreich [Gol3], our definitions of zero knowledge (in Section 2.5) refer to simulators that run in strict polynomial time. In this section, we extend our results to the the original Goldwasser–Micali–Rackoff [GMR] definition, which allowed the simulator to run in expected polynomial time. Indeed, we will prove that the two definitions yield exactly the same class **ZK**; that is, every problem having a zero-knowledge proof with an expected polynomial-time simulator also has one with a strict polynomial-time simulator. In fact, we will consider a further relaxation, captured by the following definitions.

Definition 7.15 *For a function $\varepsilon : \mathbb{N} \to [0,1]$, we say that two auxiliary-input probability ensembles $\{X_x\}$ and $\{Y_x\}$ are* ε-indistinguishable *on $I \subseteq \{0,1\}^*$ if for every nonuniform PPT D, there exists a negligible function μ such that for all $x \in I$,*

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

Definition 7.16 (weak zero knowledge) *An interactive proof system (P, V) for a promise problem Π is* weak honest-verifier zero knowledge *if for every polynomial p, there exists a probabilistic (strict) polynomial-time* simulator *S such that the ensembles $\{\langle P, V\rangle(x)\}_{x \in \Pi_Y}$ and $\{S(x)\}_{x \in \Pi_Y}$ are $(1/p(n))$-indistinguishable.*

weak-HVZK *denotes the class of promise problems having weak honest-verifier zero-knowledge proofs, respectively.*

The above definition is more relaxed than allowing expected polynomial-time simulators, because if a simulator S has expected running time $t(n)$, then running it for $p(n) \cdot t(n)$ steps yields a strict polynomial-time simulator whose output distribution is $(1/p(n))$-close to that of S. In particular, if the verifier's view is computationally indistinguishable from the output of S, then it is $(1/p(n))$-indistinguishable from the truncated version of S.

In this section, we will prove:

Theorem 7.17 **weak-HVZK = ZK**.

Analogous results were previously known for statistical zero knowledge [GV] and non-interactive statistical zero knowledge [GSV2].

By the definitions, **ZK** $\subseteq$ **weak-HVZK**, so we need only show **weak-HVZK** $\subseteq$ **ZK**. We will do this by showing that every problem in **weak-HVZK** satisfies the SZK/OWF Characterization, and applying Theorem 6.1. (By definition, **weak-HVZK** $\subseteq$ **IP**.) We will do this by an extension of our proof that every problem in **HVZK** satisfies the SZK/OWF Characterization (from Section 3). Intuitively, the "weak" computational indistinguishability in the definition of **weak-HVZK** will translate to obtaining a "weak" one-way function (in the sense that the inversion probability is bounded by, say, 1/2 rather than being negligible), and then we will apply the Yao's conversion from weak one-way functions to standard one-way functions (see [Gol3, Thm. 2.3.2]).

We begin with an extension of Lemma 3.6.

Lemma 7.18 *If a promise problem Π is in* **HVZK**, *then Π satisfies the following* Weak Conditional Pseudoentropy Characterization*: there exists a fixed polynomial m such that for every polynomial p, there is a polynomial-time computable function mapping strings x to a* samplable joint distribution *(X, Y) on $\{0,1\}^{m(|x|)} \times \{0,1\}^{m(|x|)}$ and a parameter r such that*

- *If $x \in \Pi_Y$, then there exists a (not necessarily samplable) joint distribution (X', Y') such that (X', Y') is $(1/p(n))$-indistinguishable from (X, Y) and $\mathrm{H}(X'|Y') \geq r$, and*
- *If $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 1$,*

A crucial point is that the output length m of the circuits X and Y does not grow with the level of indistinguishability required (as specified by p). However, the sizes of the circuits and their input length can indeed depend on p.

Proof Sketch: Recall that the proof of Lemma 3.6 first constructed distributions X, Y as follows:

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

where $\ell = \ell(|x|)$ is the number of rounds in the proof system. Here we do the same, but take S to be the simulator achieving ε-indistinguishability, where $\varepsilon = 1/(\ell(|x|) \cdot p(|x|))$ and p is any given polynomial.

As in the proof of Lemma 3.6, when $x \in \Pi_Y$, then (X, Y) is ε-indistinguishable from $(X', Y') = (\langle P, V\rangle_{2I}, \langle P, V\rangle_{2I-1})$, where I denotes a uniform random element of $\{1, \ldots, \ell\}$, and $H(X'|Y') = r$. And when $x \in \Pi_N$, then $H(X|Y) \leq (r-1)/\ell$.

Then final distributions are taken to be $((X_1, \ldots, X_\ell), (Y_1, \ldots, Y_\ell))$ where each (X_i, Y_i) is an independent copy of (X, Y). This increases the entropy gap to 1 bit as before, and the level of indistinguishability deteriorates to $(\ell \cdot \varepsilon) < 1/p(|x|)$. Notice that the output lengths of these distributions depend only on the communication complexity of the proof system (but the circuit sizes and number of random bits required depend on the simulator, which in turn may depend on the choice of p). □

Given this lemma, we proceed to extend the reduction from the Conditional Pseudoentropy Characterization to the SZK/OWF Characterization.

Lemma 7.19 *If a promise problem satisfies the* Weak Conditional Pseudoentropy Characterization*, then it satisfies the* SZK/OWF Characterization*.*

Proof: Given an instance x of the promise problem Π, for any $\varepsilon = \varepsilon(n) = 1/\mathrm{poly}(n)$, we can efficiently construct two samplable distributions (X, Y) on $\{0,1\}^m \times \{0,1\}^m$ and parameter r such that if $x \in \Pi_Y$, then $\mathrm{H}(X'|Y') \geq r + 2$ for some (X', Y') that is ε-indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 2$. Again, $m = m(|x|)$ is a fixed polynomial independent of ε.

Let I be the set of instances $x \in \Pi_Y$ such that $\mathrm{H}(X|Y) < r$. The proof that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in **SZK** is identical the proof of Lemma 3.9.

Thus, we focus on constructing one-way functions on I. The first step of the construction does not change. We set $k = 4n \cdot (m+n)^2$, and consider the samplable distributions

$$\begin{aligned} Z &= (H, Y_1, \ldots, Y_k, H(X_1, \ldots, X_k)), \text{ and} \\ Z' &= (H, Y_1, \ldots, Y_k, U_{kr+1}), \end{aligned}$$

As in the proof of Lemma 3.9, $H(Z') \geq \mathrm{H}(Z) + 1$. The only change is that instead of arguing that Z and Z' are computationally indistinguishable, we claim that that they are ε'-indistinguishable from Z for $\varepsilon' = k \cdot \varepsilon$. (The deterioration by a factor of k comes from taking k samples of (X_i, Y_i).)

Recalling that $k = 4n \cdot (m+n)^2$ depends only on n and the output length m, we see that we can still make the level ε' of indistinguishability arbitrarily small. Moreover, the output length m' of Z and Z' remain independent of the choice of $\varepsilon' = 1/\mathrm{poly}(n)$.

Now, to obtain a (weak) one-way function, we perform one more round of flattening and hashing. This is essentially the construction of Håstad et al. [HILL] going from a "false entropy generator" to a "pseudoentropy generator" — where the output is indistinguishable from a distribution whose min-entropy is higher than the seed-length of the generator. However, since we are starting from only a weak false entropy generator Z as above, we need to ensure that the level of indistinguishability deteriorates only as a function of the output length m' of Z and the security parameter (and not the input length).

This part of the construction depends on "guess" e for (an approximation to) the entropy of Z. (At the end we will enumerate over all choices for e.) Specifically, set $k' = 2n(m'+n)^2$, let q be the number of input gates to Z (as a circuit), let G be a random universal hash function mapping $\{0,1\}^{k'q}$ to $\{0,1\}^{k'q-k'e-n}$, and consider the following samplable distributions:

$$\begin{aligned} W_e &= (Z(R_1), \ldots, Z(R_{k'}), G, G(R_1, \ldots, R_{k'})), \text{ and} \\ W'_e &= (Z'_1, \ldots, Z'_k, G, U_{k'q-k'e-k'/8}), \end{aligned}$$

where $R_1, \ldots, R_{k'}$ are independent copies of U_q, and $Z'_1, \ldots, Z'_k$ are independent copies of Z'.

Claim 7.20 *For* $\mathrm{H}(Z) \leq e \leq \mathrm{H}(Z) + 1/2$*, we have:*

1. *W_e and W'_e are $k'\varepsilon'$-indistinguishable.*
2. $\Pr[W'_e \in \mathrm{Supp}(W_e)] \leq (k'+2) \cdot 2^{-n}$.

Before proving the claim, we describe how it completes the proof of the Lemma. Specifically, we argue that the circuit generating W_e defines a (weak) one-way function. Any algorithm that inverts W_e with probability at least δ can be used to distinguish between W_e and W'_e with advantage at least $\delta - 2^{-n}$ (because by Item 2 it is information-theoretically impossible to find a W_e-preimage of a random sample of W'_e, except with probability $(k'+2) \cdot 2^{-n}$). By Item 1, we conclude that W_e can be inverted with probability at most $\delta = k'\varepsilon' + (k'+2) \cdot 2^{-n} \leq 1/2$, and is thus a weak one-way function. Since we do not know the value of $\mathrm{H}(Z)$, we consider the function $f_x(r_1, \ldots, r_{2m'}) = (W_{1/2}(r_1), W_1(r_2), \ldots, W_{m'-1/2}(r_{2m'-1}), W_{m'}(r_{2m'}))$, which is a weak one-way function because one of its components is a weak one-way function (and the others are independent). Applying the standard reduction from weak one-way functions to standard one-way functions completes the proof. Thus, all that remains is to establish Claim 7.20.

> **Proof of claim:** It will first be useful to remove low-probability samples from both Z and Z'. Let
>
> $$L = \{z : \Pr[Z = z] \leq 2^{-n} \cdot 2^{-m'}\}.$$
>
> By a union bound, $\Pr[Z \in L] \leq 2^{-n}$. Then $\hat{Z} = Z|_{Z \notin L}$ is 2^{-n}-close to Z and moreover for every $z \in \mathrm{Supp}(\hat{Z})$,
>
> $$\Pr\left[\hat{Z} = z\right] \geq \Pr[Z = z] \geq 1/2^{m'+n}.$$

By Lemma 2.1, we have $|\mathrm{H}(\hat{Z}) - \mathrm{H}(Z)| = 2^{-n} \cdot m' + \mathrm{H}_2(2^{-n})$, which is negligible. By the Flattening Lemma, $\otimes^{k'} \hat{Z}$ is Δ-flat for $\Delta = \sqrt{k'} \cdot (m' + n)$ Analogously, using Z' we can define L', and $\hat{Z}'$, and draw the same conclusions.

The Δ-flatness of $\otimes^{k'} \hat{Z}$ implies that with probability at least $1 - 2^{-n}$ over $\overline{z} = (z_1, \ldots, z_k) getsr \otimes^{k'} \hat{Z}$, we have

$$\Pr[\otimes^{k'} \hat{Z} = \overline{z}] \geq 2^{-\sqrt{n} \cdot \Delta} \cdot 2^{-k' \cdot \mathrm{H}(\hat{Z})}.$$

Since $\otimes^{k'} Z$ and $\otimes^{k'} \hat{Z}$ are $k' \cdot 2^{-n}$-close (by Lemma 2.2), the same holds with probability at least $1 - (k'+1) \cdot 2^{-n}$ over $\overline{z} \leftarrow \otimes^{k'} Z$. For any such $\overline{z}$, we have

$$\begin{aligned}
\#\{(r_1, \ldots, r_k) : \forall i \ Z(r_i) = z_i\} &= 2^{k'q} \cdot \Pr[\otimes^{k'} Z = \overline{z}] \\
&\geq 2^{k'q} \cdot \Pr[\otimes^{k'} Z = \overline{z} | \otimes^{k'} Z \in (L^c)^{k'}] \cdot \Pr\left[\otimes^{k'} Z \in (L^c)^{k'}\right] \\
&\geq 2^{k'q} \cdot \Pr[\otimes^{k'} \hat{Z} = \overline{z}] \cdot (1 - k' \cdot 2^{-n}) \\
&\geq 2^{k'q} \cdot 2^{-\sqrt{n} \cdot \Delta - k' \cdot \mathrm{H}(\hat{Z})} \cdot (1 - k' \cdot 2^{-n}) \\
&\geq 2^{k'q - k'e - k'/8 + 2n},
\end{aligned}$$

where in the last inequality we use the fact that $\mathrm{H}(\hat{Z}) \geq \mathrm{H}(Z) - \mathrm{neg}(n) \geq e - \mathrm{neg}(n)$ and $\sqrt{n} \cdot \Delta \leq k'/16$, $2n + 1 \leq k'/16$. for sufficiently large n. This implies that conditioned on $(Z(R_1), \cdots, Z(R_{k'})) = \overline{z}$, the min-entropy of $(R_1, \ldots, R_{k'})$ is at least $k'q - k'e - k'/8 + 2n$. Thus, by the Leftover Hash Lemma 2.6, $(G, G(R_1, \ldots, R_k))$ is 2^{-n}-close to $(G, U_{k'q - k'e - k'/8})$. We conclude that W_e is statistically indistinguishable from

$$V = (Z_1, \ldots, Z_k, G, U_{k'q - k'e - k'/8}),$$

where $Z_1, \ldots, Z_k$ are independent copies of Z. Since Z is ε'-indistinguishable from Z', it follows that V is $(k'\varepsilon')$-indistinguishable from W'_e. Therefore, W_e and W'_e are $(k'\varepsilon')$-indistinguishable, as desired.

Now we proceed to Item 2. First we bound $|\mathrm{Supp}(W_e)|$. Let g be the number of random bits to generate G. Then the number of random bits used to generate W_e is at most $k'q + g$. Hence $|\mathrm{Supp}(W_e)| \leq 2^{k'q+g}$. Next show that W'_e is statistically indistinguishable from a distribution with min-entropy significantly higher than $k'q + g$. This amounts to lower-bounding the min-entropy of $(Z'_1, \ldots, Z'_k) = \otimes^{k'} Z'$, since the remaining components of the W'_e are independent and have min-entropy $g + k'q - k'e - k'/8$. As above, instead of Z', we consider $\hat{Z}'$. Recall that $\otimes^{k'} \hat{Z}'$ is $(k'2^{-n})$-close to $\otimes^k Z'$, and is Δ-flat. By Δ-flatness, $\otimes^{k'} \hat{Z}'$ is 2^{-n}-close to a distribution with min-entropy $k' \cdot \mathrm{H}(\hat{Z}') - \sqrt{n}\Delta \geq k' \cdot (e + 1/2) - k'/4 = k'e + k'/4$ for sufficiently large n. Therefore, W'_e is $(k'+1) \cdot 2^{-n}$-close to a distribution with min-entropy at least

$$(g + k'q - k'e - k'/2) + (k'e + k'/4) \geq k'q + g + n.$$

A distribution of min-entropy $w = k'q + g + n$ can land in $\mathrm{Supp}(W_e)$ with probability at most $2^{-w} \cdot |\mathrm{Supp}(W_e)| \leq 2^{-n}$. Therefore W'_e lands in $\mathrm{Supp}(W_e)$ with probability at most $2^{-n} + (k'+1) \cdot 2^{-n}$, as desired. □

■

8 Open Problems

There are some results that are known about **ZK** under the assumption that one-way functions exist, but we do not know how to prove unconditionally:

- **ZK** is closed under complement. (If one-way functions exist, then **ZK** = **PSPACE** = **co-PSPACE**.)
- If $\Pi \in \mathbf{ZK} \cap \mathbf{NP}$, then Π has a constant-round zero-knowledge proof (with soundness error $1/\mathrm{poly}(n)$) [GMW, Blu]. (Constant-round protocols with negligible soundness error are known under stronger assumptions [GK1].)
- If $\Pi \in \mathbf{ZK} \cap \mathbf{NP}$, then Π has a computational zero-knowledge proof where the prover runs in probabilistic polynomial time given a witness for membership. (In our Theorem 6.1, the prover needs an **NP** *oracle*.)

The only bottleneck for proving the latter two results unconditionally is our problem-dependent commitment scheme for **SZK** (Theorem 4.3) so any improvement to that commitment scheme with respect to round complexity or prover efficiency would have an analogous impact on **ZK**. (And on **SZK** — indeed, in [MV], problem-dependent commitments were proposed as an approach to proving the **SZK**-analogue of the last item.)

A natural next project is to undertake a similar unconditional study of zero-knowledge *arguments*, but there are several obstacles that need to be overcome. For example, the notion of argument systems is less meaningful if the specified prover is not constrained to run in polynomial time, but our techniques currently only give $\mathbf{BPP}^{\mathbf{NP}}$ provers (due to the problem-dependent commitment for **SZK**).

Acknowledgments

I am grateful to Emanuele Viola for an inspiring conversation about pseudoentropy and [HILL] that prompted me to revisit the questions addressed in this paper. I thank Oded Goldreich, Shafi Goldwasser, Shien Jin Ong, and the anonymous reviewers for clarifying discussions and help in improving the presentation. I also thank Danny Gutfreund, Madhu Sudan, and Luca Trevisan for some old conversations that have influenced this work.

References

[AH] W. Aiello and J. Håstad. Statistical Zero-Knowledge Languages Can Be Recognized in Two Rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, June 1991.

[BM] L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System and a Hierarchy of Complexity Classes. *J. Comput. Syst. Sci.*, 36:254–276, 1988.

[Bar] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. *Proc. of the 42nd FOCS, 2001*, 2001. Preliminary full version available on `http://www.wisdom.weizmann.ac.il/~boaz`.

[BLV] B. Barak, Y. Lindell, and S. Vadhan. Lower Bounds for Non-Black-Box Zero Knowledge. Technical Report TR04–083, Electronic Colloquium on Computational Complexity, September 2004. Extended abstract in *FOCS '04*.

[BGP] M. Bellare, O. Goldreich, and E. Petrank. Uniform Generation of NP-Witnesses Using an NP-Oracle. *Information and Computation*, 163, 2000.

[BMO] M. Bellare, S. Micali, and R. Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 482–493, Baltimore, Maryland, 14–16 May 1990.

[BGG$^+$] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Everything Provable is Provable in Zero-Knowledge. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer-Verlag, 1990, 21–25 Aug. 1988.

[BBR] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988. Special issue on cryptography.

[Blu] M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Berkeley, Calif., 1986)*, pages 1444–1451, Providence, RI, 1987. Amer. Math. Soc.

[BCC] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, Oct. 1988.

[CT] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, Inc., 2nd edition, 1991.

[DDPY] A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung. Image Density is Complete for Non-interactive-SZK. In *Automata, Languages and Programming, 25th International Colloquium*, Lecture Notes in Computer Science, pages 784–795, Aalborg, Denmark, 13–17 July 1998. Springer-Verlag. See also preliminary draft of full version, May 1999.

[DOY] G. Di Crescenzo, T. Okamoto, and M. Yung. Keeping the SZK-Verifier Honest Unconditionally. In B. S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 31–45. Springer-Verlag, 17–21 Aug. 1997.

[ESY] S. Even, A. L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Information and Control*, 61(2):159–173, May 1984.

[For] L. Fortnow. The Complexity of Perfect Zero-Knowledge. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.

[FGM$^+$] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On Completeness and Soundness in Interactive Proof Systems. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 429–442. JAC Press, Inc., 1989.

[Gol1] O. Goldreich. A note on computational indistinguishability. *Inf. Process. Lett.*, 34(6):277–281, 28 May 1990.

[Gol2] O. Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol3] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[Gol4] O. Goldreich. On Promise Problems (a survey in memory of Shimon Even [1935-2004]). Technical Report TR05–018, Electronic Colloquium on Computational Complexity, February 2005.

[GK1] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GK2] O. Goldreich and H. Krawczyk. Sparse pseudorandom distributions. *Random Structures & Algorithms*, 3(2):163–174, 1992.

[GK3] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM J. Comput.*, 25(1):169–192, Feb. 1996.

[GK4] O. Goldreich and E. Kushilevitz. A Perfect Zero-Knowledge Proof System for a Problem Equivalent to the Discrete Logarithm. *Journal of Cryptology*, 6:97–116, 1993.

[GMW] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *J. ACM*, 38(1):691–729, 1991.

[GO] O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology*, 7(1):1–32, Winter 1994.

[GP] O. Goldreich and E. Petrank. Quantifying knowledge complexity. *Computational Complexity*, 8(1):50–98, 1999.

[GSV1] O. Goldreich, A. Sahai, and S. Vadhan. Honest Verifier Statistical Zero-Knowledge Equals General Statistical Zero-Knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, Dallas, 23–26 May 1998.

[GSV2] O. Goldreich, A. Sahai, and S. Vadhan. Can Statistical Zero-Knowledge be Made Non-Interactive?, or On the Relationship of SZK and NISZK. In *Advances in Cryptology—CRYPTO '99*, Lecture Notes in Computer Science. Springer-Verlag, 1999, 15–19 Aug. 1999. To appear.

[GV] O. Goldreich and S. Vadhan. Comparing Entropies in Statistical Zero-Knowledge with Applications to the Structure of SZK. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 54–73, Atlanta, GA, May 1999. IEEE Computer Society Press.

[GMR] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, 18(1):186–208, February 1989.

[GS] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 73–90. JAC Press, Inc., 1989.

[HILL] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396 (electronic), 1999.

[ILL] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random Generation from one-way functions (Extended Abstracts). pages 12–24.

[IY] R. Impagliazzo and M. Yung. Direct Minimum-Knowledge Computations (Extended Abstract). In C. Pomerance, editor, *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer-Verlag, 1988, 16–20 Aug. 1987.

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

[JVV] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Comput. Sci.*, 43(2-3):169–188, 1986.

[LLS] R. E. Ladner, N. A. Lynch, and A. L. Selman. A Comparison of Polynomial Time Reducibilities. *Theoretical Comput. Sci.*, 1(2):103–123, Dec. 1975.

[LFKN] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic Methods for Interactive Proof Systems. *J. ACM*, 39(4):859–868, Oct. 1992.

[MV] D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In D. Boneh, editor, *Advances in Cryptology—CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer-Verlag, 17–21 August 2003.

[Nao] M. Naor. Bit Commitment Using Pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[NZ] N. Nisan and D. Zuckerman. Randomness is Linear in Space. *J. Comput. Syst. Sci.*, 52(1):43–52, Feb. 1996.

[Oka] T. Okamoto. On Relationships Between Statistical Zero-Knowledge Proofs. *Journal of Computer and System Sciences*, 60(1):47–108, February 2000.

[Ost] R. Ostrovsky. One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 133–138, Chicago, Illinois, 30 June–3 July 1991. IEEE Computer Society Press,.

[OW] R. Ostrovsky and A. Wigderson. One-Way Functions are Essential for Non-Trivial Zero-Knowledge. In *Proceedings of the Second Israel Symposium on Theory of Computing and Systems*, 1993.

[PT] E. Petrank and G. Tardos. On the Knowledge Complexity of $\mathcal{NP}$. In *37th Annual Symposium on Foundations of Computer Science*, pages 494–503, Burlington, Vermont, 14–16 Oct. 1996. IEEE.

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

[Sha] A. Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, Oct. 1992.

[Vad1] S. P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs.* PhD thesis, Massachusetts Institute of Technology, August 1999.

[Vad2] S. P. Vadhan. An Unconditional Study of Computational Zero Knowledge. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04)*, pages 176–185, Rome, Italy, 17–19 October 2004.

A Proof of the Flattening Lemma

Lemma A.1 (Flattening Lemma, restated) *Let X be a distribution, k a positive integer, and $\otimes^k X$ denote the distribution composed of k independent copies of X. Suppose that for all x in the support of X it holds that $\Pr[X = x] \geq 2^{-m}$. Then $\otimes^k X$ is $\sqrt{k} \cdot m$-flat.*

Suppose Y is jointly distributed with X, and for all (x, y) in the support of (X, Y) it holds that $\Pr[X = x|Y = y] \geq 2^{-m}$. Then, defining $((X_1, Y_1), \ldots, (X_k, Y_k)) = \otimes^k (X, Y)$, the random variable $(X_1, \ldots, X_k)$ is $\sqrt{k} \cdot m$-flat given $(Y_1, \ldots, Y_k)$.

Proof: For every (x, y) in the support of (X, Y), we define the *weight of x given y* to be $\mathrm{wt}(x|y) = \log(1/\Pr[X = x|Y = y])$. Then $\mathrm{wt}(\cdot)$ maps the support of (X, Y) to $[0, m]$. For every $x_1, \ldots, x_k, y_1, \ldots, y_k$, we have

$$\log \frac{1}{\Pr[(X_1, \ldots, X_k) = (x_1, \ldots, x_k)|(Y_1, \ldots, Y_k) = (y_1, \ldots, y_k)]} = \sum_{i=1}^{k} \mathrm{wt}(x_i|y_i).$$

Thus, if we let $\overline{X} = (X_1, \ldots, X_k)$, $\overline{Y} = (Y_1, \ldots, Y_k)$, we have:

$$\Pr\left[\overline{X} \text{ is not } t\Delta\text{-typical given } \overline{Y}\right] = \Pr\left[\left|\sum_{i=1}^{k} \mathrm{wt}(X_i|Y_i) - \mathrm{H}(\overline{X}|\overline{Y})\right| \geq t\Delta\right].$$

For every i, $\mathrm{E}[\mathrm{wt}(X_i|Y_i)] = \mathrm{H}(X|Y)$ and $\mathrm{H}(\overline{X}|\overline{Y}) = k \cdot \mathrm{H}(X|Y)$, so we are bounding the probability that the average of k independent, identically distributed random variables taking values in $[0, m]$ deviates from its expectation by $t\Delta/k$. By the Hoeffding Inequality, this probability is at most

$$2 \cdot \exp\left(\frac{-2 \cdot k \cdot (t\Delta/k)^2}{m^2}\right).$$

For $\Delta = \sqrt{k} \cdot m$ and $t \geq 1$, this bound becomes $2\exp(-2t^2) \leq 2^{-t^2}$, establishing the lemma. ■