Electronic Colloquium on Computational Complexity, Report No. 56 (2006)

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/

April 27, 2006

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.
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 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

[*] A preliminary version of this paper appeared in *FOCS '04* [Vad2] and another version will appear in the *SIAM Journal on Computing* [Vad3].

[†] 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.

ISSN 1433-8092

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, and 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]

- Can we transform proof systems that are zero knowledge for the "honest verifier" (*i.e.,* the verifier that follows the specified protocol) into ones that are zero knowledge in general (*i.e.,* 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**, where **IP** is the class of problems possessing interactive proofs [GMW, IY, BGG$^+$, Nao, HILL]. Thus, **ZK** is completely characterized and moreover has natural complete problems (namely, any complete problem for **IP** = **PSPACE** [LFKN, Sha2]). This also implies that **HVZK** equals **ZK**, since **ZK** ⊆ **HVZK** ⊆ **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 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 quite complete problems, but turn out to have much of the same utility.).

- Prove that **HVZK** = **ZK**.

[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 [BCC], 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 multiplicative factor suffices (by sequential repetition).

- Show how to transform any computational zero knowledge proof into one with public coins and perfect completeness.
- Establish closure properties of **ZK**, such as closure under union.

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.,* $\mathbf{ZK} \neq \mathbf{BPP}$), then "some form of one-way functions" exist. Then they made the appealing suggestion that one might prove unconditional results about computational zero knowledge by a case analysis: If $\mathbf{ZK} = \mathbf{BPP}$, 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 verifier decides membership on its own using the **BPP** algorithm). On the other hand, if $\mathbf{ZK} \neq \mathbf{BPP}$, 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 $\mathbf{ZK} \neq \mathbf{BPP}$) are too weak for the conditional constructions mentioned above.[3] (See Section 7.1 for more details.)

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 $\mathbf{ZK} \neq \mathbf{SZK}$, 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 $\mathbf{ZK} = \mathbf{SZK}$ 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, 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

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

[4] Recall that the zero-knowledge property is formalized by requiring that there be a 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.

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]. (See [Vad1] for a unified presentation of all of these results.) 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 [GV, 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*[5] $\Pi = (\Pi_Y, \Pi_N)$ *satisfies the* SZK/OWF Condition *if there exists a set $I \subseteq \Pi_Y$ of* yes *instances, a polynomial-time computable function f, 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 $\Pi' \in$ **SZK**, where $\Pi' = (\Pi_Y \setminus I, \Pi_N)$.*
- *When $x \in I$, the function $f_x(\cdot) \stackrel{\text{def}}{=} f(x, \cdot)$ is hard to invert. That is, for every nonuniform polynomial-time algorithm A, there exists a negligible function ϵ such that for every $x \in I$,*

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

Intuitively, this characterization says that for every yes instance x, either one can prove the membership of x in Π_Y 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, *i.e.,* without auxiliary input), then *all* promise problems satisfy the SZK/OWF Condition (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, the above condition (regarding Π) alone *cannot* characterize **ZK**, since if one-way functions do exist Π will satisfy Definition 1.1 even if $\Pi \notin$ **IP**. We prove that if we simply add the condition $\Pi \in$ **IP**, then we do indeed obtain an exact characterization.

Theorem 1.2 (SZK/OWF Characterization of ZK)
$\Pi \in$ **ZK** *if and only if* $\Pi \in$ **IP** *and* Π *satisfies the* SZK/OWF Condition.

[5] A promise problem Π consists of a pair (Π_Y, Π_N) of disjoint sets of strings, corresponding to the yes instances and no instances of Π, respectively [ESY]. All of the complexity classes we consider in this paper, e.g. **ZK**, **SZK**, and **IP**, are taken to be classes of promise problems. See Section 2.3.

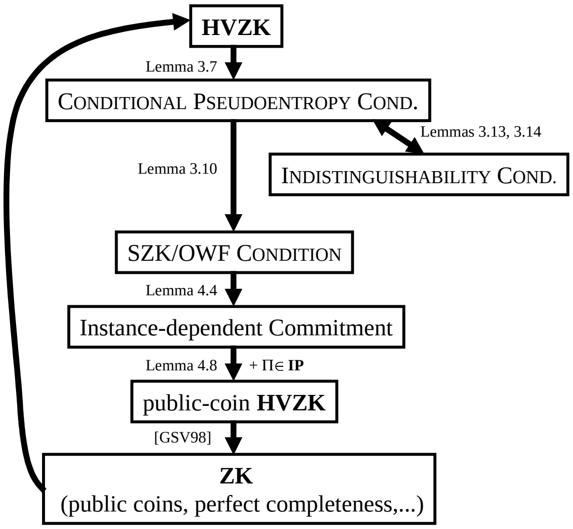
HVZK
Lemma 3.7
CONDITIONAL PSEUDOENTROPY COND.
Lemmas 3.13, 3.14
INDISTINGUISHABILITY COND.
Lemma 3.10
SZK/OWF CONDITION
Lemma 4.4
Instance-dependent Commitment
Lemma 4.8
+ Π∈ IP
public-coin HVZK
[GSV98]
ZK
(public coins, perfect completeness,...)

Figure 1: Steps of our proof

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. When 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 CONDITION. In the reverse ("if") direction, we show that every problem in **IP** satisfying the SZK/OWF CONDITION 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.

1.2 Proof Outline

Figure 1 illustrates our main steps in establishing both directions of Theorem 1.2. In proving the forward direction, we first prove that every problem in **HVZK** satisfies a "CONDITIONAL PSEUDOENTROPY CONDITION" and a "INDISTINGUISHABILITY CONDITION." These are computational analogues of the known complete problems for **SZK** [SV, GV], and are described in more detail below. The reductions from **HVZK** to these characterizations are natural 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]). We then show that every problem satisfying the CONDITIONAL PSEUDOENTROPY CONDITION also satisfies the SZK/OWF CONDITION. This step utilizes the techniques of Håstad, Impagliazzo, Levin, and Luby [HILL] to construct the needed one-way functions f_x.

In proving the reverse direction of Theorem 1.2, we first show that every problem satisfying the SZK/OWF CONDITION has a certain kind of "instance-dependent commitment scheme" [IOS] as discussed in more detail below. We then use the techniques of [GMW, IY, BGG$^+$, IOS] to show that every problem in **IP** with such a commitment scheme has a public-coin honest-verifier zero-knowledege proof. The honest-verifier zero-knowledge proof is then converted into one that is zero knowledge even for cheating verifiers using the compiler of [GSV1]. The resulting proof system remains public coin, and also has additional nice properties such as perfect completeness and black-box simulation.

Putting these steps together, we deduce that membership in **HVZK**, membership in **ZK** (even with additional nice properties), the SZK/OWF CONDITION, the CONDITIONAL PSEUDOENTROPY CONDITION, the INDISTINGUISHABILITY CONDITION, and having an instance-dependent commitment scheme are all equivalent (for problems in **IP**). The latter three characterizations of **ZK** are of interest beyond their role in establishing Theorem 1.2, so we describe them in more detail below.

1.3 Additional Characterizations of ZK

Computational Analogues of the SZK-complete problems. Recall that in [SV, GV], it was demonstrated that **SZK** has two natural complete problems, STATISTICAL DIFFERENCE and ENTROPY DIFFERENCE. These problems proved to be a very useful tool in the study of **SZK** (cf., [SV, Vad1]), because they reduced the study of the entire class to the study of these specific problems.

In this work, we provide characterizations of **ZK** that are natural computational analogues of these two problems. For example, recall that instances of the STATISTICAL DIFFERENCE problem consist of pairs (X, Y) of probability distributions on strings, specified by circuits that sample from them. The YES instances are pairs that are statistically close and the NO instances are pairs that are statistically far apart. We show that if "statistically close" is replaced with "computationally indistinguishable," the condition that results characterizes **ZK**. This condition, which we refer to as the INDISTINGUISHABILITY CONDITION, cannot be cast a complete problem because there can be distributions that are both computationally indistinguishable and statistically far apart. Rather, we say that a promise problem satisfies the INDISTINGUISHABILITY CONDITION if its instances can be efficiently mapped to pairs (X, Y) that are computationally indistinguishable or statistically far apart, according to whether the instance is a YES or NO instance. We show that this characterizes **ZK** in the sense that a promise problem is in **ZK** iff it is in **IP** and satisfies the INDISTINGUISHABILITY CONDITION.

The computational analogue of ENTROPY DIFFERENCE is less immediate, and in fact a crucial step towards our establishment of the SZK/OWF Characterization Theorem was the realization that the "right" problem to generalize is a variant of ENTROPY DIFFERENCE, which we call CONDITIONAL ENTROPY APPROXIMATION, rather than ENTROPY DIFFERENCE itself. (See Section 3 for more details.)

Instance-dependent commitments. A fundamental tool in the construction of many zero-knowledge proofs is that of a *commitment scheme*. This is a protocol whereby a *sender* can 'commit' to a bit b in such a way that the *receiver* learns nothing about b (the scheme is *hiding*), but nevertheless the sender cannot 'open' the commitment to a value other than b (the scheme is *binding*). Commitment schemes, which can be constructed from any one-way function [Nao, HILL], play an essential role in the construction of zero-knowledge proofs for all of **NP** and **IP** [GMW, IY, BGG$^+$]. Some evidence that commitments are necessary for zero knowledge came from the work of Damgård [Dam1, Dam2], who focused on 3-round public-coin zero-knowledge proofs, and

Ostrovsky and Wigderson [Ost, OW], who showed that zero-knowledge proofs for *hard-on-average* languages imply one-way functions (and hence standard commitment schemes [Nao, HILL]).

In this work, we show an *equivalence* between zero-knowledge proofs and certain types of commitment schemes, which we now describe. In an *instance-dependent* commitment scheme [BMO, IOS, MV] for a promise problem Π, both the sender and receiver get an common auxiliary input x, which is an instance of Π. It is required that if x is a YES instance of Π, then the scheme is hiding, and if x is a NO instance, then the scheme is binding. Thus, instance-dependent commitment schemes 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 a promise problem is in **ZK** (resp., **SZK**) if and only if it has an instance-dependent commitment scheme that is computationally (resp., statistically) hiding on YES instances (and statistically binding on NO instances). Indeed, the most technical part of this paper is the construction of instance-dependent commitment schemes for all of **SZK**, which utilizes much of the machinery previously developed in the study of **SZK** [Oka, SV, GV]. The construction of instance-dependent commitments for **ZK** then follows using the SZK/OWF Characterization Theorem and the known construction of commitment schemes from one-way functions [Nao, HILL].

Two deficiencies in our instance-dependent commitments are that the hiding property only holds against an honest receiver (*i.e.,* one that follows the specified protocol), and that the sender of the commitment scheme is not polynomial time, but rather $\mathbf{BPP}^{\mathbf{NP}}$. The effect of these are that the direct constructions of zero-knowledge proofs that we obtain using the commitments are only honest-verifier zero knowledge and have provers that require an **NP** oracle (rather than just an **NP** witness, as would be preferable for problems in **NP**). The honest-verifier constraint is removed using the compiler of [GSV1], which converts *public-coin* honest-verifier zero-knowledge proofs into general zero-knowledge proofs. The **NP** oracle has been removed in subsequent work [NV], by using a new, more relaxed, type of instance-dependent commitment scheme; see Section 8.

1.4 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 instance-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 Theorems [OW], 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 the pair (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, $\mathrm{poly}(n)$ denotes an arbitrary function $f(n) = n^{O(1)}$.

For a probabilistic algorithm A, we write $A(x; r)$ to denote the output of A on input x and coin tosses r. In this case, $A(x)$ is a random variable representing 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 sequence of strings where $|z_n| = \mathrm{poly}(n)$, and A is a PPT algorithm that receives pairs of inputs of the form $(x, z_{|x|})$. (The string z_n is the called the *advice string* for A for inputs of length n.) Nonuniform PPT algorithms are equivalent to (nonuniform) families of polynomial-sized Boolean circuits.

A boolean 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 C's input gates being the coin tosses; thus we will often refer to distributions specified by circuits as *samplable distributions*. This is a "nonuniform" notion of samplability, because the sampling algorithm C can be tailored to a particular output length n. Later, in Definition 2.11, we will consider a uniform notion of samplability, which refers to ensembles (*i.e.,* sequences) of probability distributions that are generated by a uniform PPT algorithm. Samplable distributions 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

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

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 to 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

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

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 $\mathcal{U}$, if $\delta = \Delta(X,Y)$, then*

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

Another useful fact is that random variables taking values in a universe $\mathcal{U}$ can be modified so that they do not assign any elements in their support a probability mass much smaller than $1/|\mathcal{U}|$, without incurring a significant statistical difference or change in entropy.

Lemma 2.2 *Let (X,Y) be a random variable taking values in $\mathcal{U}\times\mathcal{V}$. Then for any $\delta > 0$, there is a random variable (X',Y') that is δ-close to (X,Y) and satisfies* $\Pr[X'=x|Y'=y] \geq \delta/|\mathcal{U}|$ *for all* $(x,y) \in \mathrm{Supp}(X',Y')$. *Moreover,* $|\mathrm{H}(X'|Y') - \mathrm{H}(X|Y)| \leq \log(|\mathcal{U}|-1)\cdot\delta + \mathrm{H}_2(\delta)$.

Proof: For each $y \in \mathrm{Supp}(Y)$, the set $S_y = \{x \in \mathcal{U} : \Pr[X=x|Y=y] < \delta/|\mathcal{U}|\}$ has total probability mass less than δ under the conditional distribution $X|_{Y=y}$. Thus shifting the probability mass of the points in S_y to other points in $\mathcal{U}$ yields a random variable Z_y that is δ-close to $X|_{Y=y}$. By Lemma 2.1, for every y, the entropy of Z_y differs from that of $X|_{Y=y}$ by at most $\delta' = \log(|\mathcal{U}|-1)\cdot\delta + \mathrm{H}_2(\delta)$. Thus taking $(X',Y') = (Z_Y, Y)$ satisfies the conclusion of the lemma. ■

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 2.3 (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*

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

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

If Y is a random variable jointly distributed with X, and $y \in$ Supp(Y)*, we say that x is* Φ-heavy given y *(resp.,* Φ-light given y *if* $\Pr[X = x|Y = y] \geq 2^{\Phi} \cdot 2^{-\mathrm{H}(X|Y)}$ (resp., $\Pr[X = x|Y = y] \leq 2^{-\Phi} \cdot 2^{-\mathrm{H}(X|Y)}$)*. Otherwise, we say that x is* Φ-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 2.5 (nearly flat distributions) [6] *A distribution X is called* Φ-flat *if for every $t \geq 1$ the probability that an element chosen from X is $t \cdot \Phi$-typical is at least $1 - 2^{-t^2}$.*

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

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

Lemma 2.6 (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 the random variable $((X_1, Y_1), \ldots, (X_k, Y_k)) = \otimes^k(X, Y)$, 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.

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, as long as we are willing for the extraction procedure itself to be probabilistic. Surveys of the large body of work on this topic can be found in [NT, Sha1]. Much of that work focuses on minimizing the number of extra random bits used by the extractor; this is not a major concern for us, so we can use relatively simple randomness extractors. In particular, the Leftover Hash Lemma of [BBR, HILL] shows that universal (or pairwise independent) hash functions can be used for this purpose.

[6] The definition in [GV] allows any $t > 0$, but only requires that the probability of being $t\Phi$-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.

Lemma 2.7 (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 $\text{poly}(n,m)$ (cf. [Gol3, §3.6.1]).

2.3 Promise Problems

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

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

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

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

$$\begin{aligned} x \in \Pi_Y &\Rightarrow f(x) \in \Gamma_Y \\ x \in \Pi_N &\Rightarrow f(x) \in \Gamma_N. \end{aligned}$$

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 [Gol5] 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 CONDITION refers to such a variant of the notion of one-way functions (as captured the definitions below). Auxiliary-input primitives have been considered in the past, such as in the instance-dependent commitments of [IOS] (which we consider in Section 4.1) and the auxiliary-input one-way functions of [OW] (which are weaker than our formulation below). In this section, we provide a general framework for discussing and relating such primitives.

Definition 2.8 *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 2.9 *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* $x \in I$*,*

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

(We note that since A is nonuniform, it is not essential that we give it the input x, because it can be hardwired in as advice, but the definition seems more natural as above.) The standard definition of one-way function is obtained by considering $I = \{1^n : n \ge 0\}$ 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 (which they denote by $\exists 1WF$), 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 the above definition, we can restate the SZK/OWF Condition as follows.

Definition 2.10 *A promise problem* $\Pi = (\Pi_Y, \Pi_N)$ *satisfies the* SZK/OWF Condition *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 2.11 *An* auxiliary-input probability ensemble *is a collection of random variables* $\{X_x\}_{x \in \{0,1\}^*}$*, where* X_x *takes values in* $\{0,1\}^{p(|x|)}$ *for some polynomial* p*. We call such an ensemble* samplable *if there is a probabilistic polynomial-time algorithm* M *such that for every* x*, the output* $M(x)$ *is distributed according to* X_x*.*

Definition 2.12 *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* $x \in I$*,*

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

Similarly, we say that $\{X_x\}$ *and* $\{Y_x\}$ *are* statistically indistinguishable *on* $I \subseteq \{0,1\}^*$ *if the above is required for all functions* D *(instead of only nonuniform PPT). Equivalently,* $\{X_x\}$ *and* $\{Y_x\}$ *are statistically indistinguishable on* I *iff* X_x *and* Y_x *are* $\mu(|x|)$*-close for some negligible function* μ *and all* $x \in I$*. If* X_x *are* Y_x *are identically distributed for all* $x \in I$ *(i.e.,* $\mu = 0$*), 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 2.13 (cf., [Gol4, 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 2.14 *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 2.15 ([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

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 2.16 *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 $1 - c(n) > s(n) + 1/\mathrm{poly}(n)$ 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* $c = 0$*. We denote by* **IP** *the class of promise problems possessing interactive proof systems. We denote by* **AM** *the class of promise problems possessing two-round, public-coin interactive proof systems.*

AM is known to equal the class of promise problems possessing constant-round (possibly private-coin) interactive proof systems [GS, BM].

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 "honest verifier," the one that follows the specified protocol.[7]

Definition 2.17 (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* $\{\langle P, V\rangle(x)\}$ *and* $\{S(x)\}$ *are (perfectly/statistically/computationally) indistinguishable on* Π_Y*.*[8] *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 2.18 (auxiliary-input zero knowledge) [9] *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*

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

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

[8] Actually, in the case of perfect zero knowledge, it is common to allow the simulator to output a failure symbol $\perp$ with some small probability (say at most 1/2) and require that the output of $S(x)$ conditioned on non-failure is identical to $\langle P, V\rangle(x)$ (cf. [Gol4, Def. 4.3.1]). However, we are only defining perfect zero knowledge for the sake of context and will not use it anywhere else in the paper.

[9] Our formulation of auxiliary-input zero knowledge is slightly different than, but equivalent to, the definition in the textbook [Gol4]. We allow V^* to run in polynomial time in the lengths of both its input x and its auxiliary input z, but put a polynomial bound on the length of the auxiliary input. In [Gol4, Sec 4.3.3], V^* is restricted to run in time that is polynomial in just the length of the input x, and no bound is imposed on the length of the auxiliary input z (so V^* may only be able to read a prefix of z). The purpose of allowing the auxiliary input to be longer than the running time of z is to provide additional nonuniformity to the distinguisher (beyond that which the verifier has); we do this directly by allowing the distinguisher to be nonuniform in Definition 2.12.

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

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 allows one to model a priori information that the verifier may possess before the interaction begins, such as from earlier steps in a larger protocol in which the zero-knowledge proof is being used or from prior executions of the same zero-knowledge proof. As a result, auxiliary-input zero knowledge is closed under sequential composition. That is, if an auxiliary-input zero-knowledge proof is repeated polynomially many times sequentially, then it remains auxiliary-input zero knowledge [GO]. Plain zero knowledge (*i.e.,* without auxiliary inputs) is not closed under sequential composition [GK3], and thus 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 auxiliary input), and instead only requires black-box access to V^*. This type of simulation is formalized as follows.

Definition 2.19 (black-box zero knowledge) *An interactive proof system* (P, V) *for a promise problem* Π *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 V^* 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, having a constant number of rounds, and having negligible error) 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 [Gol4], 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 hold for the corresponding 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 has to hold even with respect to nonuniform distinguishers and is universally quantified over all YES instances. Uniform treatments of zero knowledge are possible (see [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) Following [Gol4], we initially 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 consider the more relaxed 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 2.12) and consequently in the formulation of zero knowledge, 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, §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. However, all of our results can be extended to 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 its 'padded' version Π' defined by $\Pi'_Y = \{(x, 1^k) : x \in \Pi_Y, k \in \mathbb{N}\}$ and $\Pi'_N = \{(x, 1^k) : x \in \Pi_N, k \in \mathbb{N}\}$. For example, our result that $\mathbf{HVZK} = \mathbf{ZK}$ implies that the security-parameterized versions of these classes are also equal: for any promise problem Π having a security-parameterized honest-verifier zero-knowledge proof, we have $\Pi' \in \mathbf{HVZK} = \mathbf{ZK}$, which implies that Π has a security-parameterized (cheating-verifier) zero-knowledge proof. Note that we are not claiming that every problem in $\mathbf{ZK}$ has a security-parameterized zero-knowledge proof. (In contrast, it was shown in [SV] that every problem in $\mathbf{SZK}$ has a security-parameterized statistical zero-knowledge proof.)

6. (Closure under reductions) All of the zero-knowledge classes defined above, in particular $\mathbf{HVZK}$ and $\mathbf{ZK}$, 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 as in Item 5 above (cf., [Vad1, Prop. 2.4.1]).

3 From HVZK to the SZK/OWF Condition

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

A first attempt. To show that every $\Pi \in$ **HVZK** satisfies the SZK/OWF Condition, 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 inputs in I, the output of the simulator and the verifier's view are statistically far apart but computationally indistinguishable. Goldreich [Gol1] has shown that 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 real 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 on $\{0,1\}^n$ (which is trivially samplable) is computationally indistinguishable from some (non-samplable) distributions that are statistically very far from uniform (indeed have entropy $\mathrm{polylog}(n)$) [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 $\Pi \in$ **SZK**.

For the second difficulty, we use the fact that a samplable distribution that is computationally indistinguishable from an arbitrary (possibly 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 Condition. 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 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 samplable 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* $\Pi \in \mathbf{SZK}$, *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 $(X,Y) \mapsto (Y,X)$).

It turns out that, for obtaining **ZK**-analogues of this Completeness Theorem, it is crucial that we do not work with ENTROPY DIFFERENCE, but with a variant, CONDITIONAL ENTROPY APPROXIMATION (CEA), defined as follows:

$$\begin{aligned}\mathrm{CEA}_Y &= \{((X,Y),r) : \mathrm{H}(X|Y) \geq r\}\\ \mathrm{CEA}_N &= \{((X,Y),r) : \mathrm{H}(X|Y) \leq r - 1\}.\end{aligned}$$

Here (X,Y) is a *samplable joint distribution* specified by two circuits that use the same coin tosses.

Proposition 3.2 CONDITIONAL ENTROPY APPROXIMATION *is complete for* **SZK**.

Proof: To show that CEA is in **SZK**, we reduce it to ED. Given an instance $((X,Y),r)$ of CEA, we define $X' = \otimes^2(X,Y)$, $Y' = (\otimes^2 Y) \otimes U_{2r-1}$. Then

$$\mathrm{H}(X') - \mathrm{H}(Y') = 2 \cdot \mathrm{H}(X,Y) - (2 \cdot \mathrm{H}(Y) + (2r-1)) = 2 \cdot (\mathrm{H}(X|Y) - r) + 1.$$

It follows that $((X,Y),r) \in \mathrm{CEA}_Y \Rightarrow (X',Y') \in \mathrm{ED}_Y$ and $((X,Y),r) \in \mathrm{CEA}_N \Rightarrow (X',Y') \in \mathrm{ED}_N$.

To show that CEA is **SZK**-hard, we provide a reduction from SD to CEA, based on the reduction from SD to ED given in [Vad1, Sec 4.4]. Given an instance (X_0, X_1) of SD, we construct the following samplable joint distribution.

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

Intuitively, both $\mathrm{H}(B|Y)$ and $\Delta(X_0, X_1)$ measure how well B can be predicted from $Y = X_B$. Indeed, it is shown in [Vad1, Claim 4.4.2] that $1 - \delta \leq \mathrm{H}(B|Y) \leq \mathrm{H}_2((1-\delta)/2)$, where $\delta = \Delta(X_0, X_1)$. Plugging in $\delta = 1/3$ and $\delta = 2/3$, we see that:

$$\begin{aligned}(X_0, X_1) \in \mathrm{SD}_Y \quad &\Rightarrow \quad \mathrm{H}(B|Y) \geq 1 - 1/3 = 2/3 \\ (X_0, X_1) \in \mathrm{SD}_N \quad &\Rightarrow \quad \mathrm{H}(B|Y) \leq \mathrm{H}_2((1-2/3)/2) < .651.\end{aligned}$$

Now we amplify the gap to more than 1 bit by taking direct products. Specifically, let $(B', Y') = ((B_1, \ldots, B_{66}), (Y_1, \ldots, Y_{66}))$, where the (B_i, Y_i)'s are independent copies of (B, Y). Then

$$\begin{aligned}(X_0, X_1) \in \mathrm{SD}_Y \quad &\Rightarrow \quad \mathrm{H}(B'|Y') \geq 66 \cdot (2/3) = 44 \\ (X_0, X_1) \in \mathrm{SD}_N \quad &\Rightarrow \quad \mathrm{H}(B'|Y') < 66 \cdot .651 < 43.\end{aligned}$$

So $(X_0, X_1) \mapsto ((B', Y'), 44)$ is a valid reduction from SD to CEA. ■

We note that the unconditional version of CEA (where Y is not present and we consider only $\mathrm{H}(X)$), called Entropy Approximation, is known to be complete for *non-interactive* statistical zero knowledge [GSV2].

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

Definition 3.3 *A promise problem Π satisfies the* Indistinguishability Condition *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 2.12),*
- *If $x \in \Pi_N$, then $\Delta(X, Y) \geq 2/3$.*

We note that the constant 2/3 in the second bullet is arbitrary. By taking direct products and applying Lemmas 2.3 and 2.13, 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 the first bullet.

Like the SZK/OWF Condition, if one-way functions exist, then every promise problem satisfies the Indistinguishability Condition: 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 are both 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**.

Theorem 3.4 (Indistinguishability Characterization of ZK)
$\Pi \in$ **ZK** *if and only if* $\Pi \in$ **IP** *and* Π *satisfies the* Indistinguishability Condition.

The preceding example, showing that every promise problem satisfies the Indistinguishability Condition if one-way functions exist, also illustrates why Π satisfying the Indistinguishability Condition 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, we expect that Indistinguishability Characterization of **ZK** will have much of the same utility as a complete problem (like STATISTICAL DIFFERENCE). Indeed, several of the results about **ZK** presented in Section 7 can be established simply by taking the corresponding proof for **SZK** and replacing STATISTICAL DIFFERENCE with the INDISTINGUISHABILITY CONDITION. However, we instead choose to use the results for **SZK** as a "black box," and reduce the **ZK** case to the **SZK** case via the SZK/OWF Characterization.

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

A characterization analogous to CONDITIONAL ENTROPY APPROXIMATION follows.

Definition 3.5 *A promise problem* Π *satisfies the* CONDITIONAL PSEUDOENTROPY CONDITION *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. It is crucial that we generalize CONDITIONAL ENTROPY APPROXIMATION instead of ENTROPY DIFFERENCE. Indeed, in [Vad1] we pointed out that the condition analogous to ENTROPY DIFFERENCE, using $\mathrm{H}(X) - \mathrm{H}(Y)$ instead of $\mathrm{H}(X|Y)$, is satisfied by *all* promise problems (regardless of whether or not one-way functions exist) and thus is useless.[10] (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 conjectures of [BLV, Sec. 9].

Theorem 3.6 (Conditional Pseudoentropy Characterization of ZK)
$\Pi \in \mathbf{ZK}$ *if and only if* $\Pi \in \mathbf{IP}$ *and* Π *satisfies the* CONDITIONAL PSEUDOENTROPY CONDITION*.*

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 CONDITION is not symmetric with respect to YES and NO instances.

Overview of the proofs of Theorems 1.2, 3.6, and 3.4. In the remainder of this section, we show that every promise problem in **HVZK** satisfies the CONDITIONAL PSEUDOENTROPY CONDITION, that the CONDITIONAL PSEUDOENTROPY CONDITION is equivalent to the INDISTINGUISHABILITY CONDITION, and that every promise problem satisfying the CONDITIONAL PSEUDOENTROPY CONDITION satisfies the SZK/OWF CONDITION. This establishes the forward ("only

[10] 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 exists a (non-samplable) distribution D of low entropy (e.g. $n/2$) that is indistinguishable from the uniform distribution U_n [GK2]. Thus, if the above characterization referred to $\mathrm{H}(X) - \mathrm{H}(Y)$, then it would hold for all promise problems, by setting $X = Y = X' = U_n$, $Y' = D$, and $r = 1$ so that $\mathrm{H}(X') - \mathrm{H}(Y') \geq n/2 \geq r$ and $\mathrm{H}(X) - \mathrm{H}(Y) = 0 = r - 1$.

if") directions of Theorems 1.2, 3.6, and 3.4. The reverse directions, showing that problems in **IP** satisfying the characterizations are in **ZK**, follow from our results in Sections 4 and 5. Section 6 puts everything together and formally deduces the theorems.

3.2 The Conditional Pseudoentropy Condition

Lemma 3.7 *If a promise problem* Π *is in* **HVZK**, *then* Π *satisfies the* Conditional Pseudoentropy Condition.

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

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

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

Claim 3.8 *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 $x \in \Pi_N$ (rather than for $x \in \Pi_Y$).

Claim 3.9 *For every* $x \in \Pi_N$*,*

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

It may seem strange to consider the simulator's output distribution on NO instances, since the zero-knowledge condition does not provide any guarantees about the quality of simulation on NO instances. Indeed, Claim 3.9 is not derived from the zero knowledge property of the proof system. Rather, it is based on the *soundness* of the proof system and the fact that the simulator always produces accepting transcripts (by our modification above). Intuitively, it says that the simulation captures at most an $s(|x|)$ fraction of the probability space of the verifier's messages. Indeed, it is shown in [AH, PT, GV] that 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 3.8, 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 3.9, 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 3.5 ■

3.3 The SZK/OWF Condition

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

Lemma 3.10 *If a promise problem satisfies the* Conditional Pseudoentropy Condition*, then it also satisfies the* SZK/OWF Condition.

The idea behind the proof is the following. If Π satisfies the Conditional Pseudoentropy Condition, 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 and 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 Conditional Entropy Approximation, which is complete for **SZK** by Proposition 3.2. 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], adapted to our auxiliary-input setting.

Definition 3.11 *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 $\mathrm{H}(F_x) \geq \mathrm{H}(D_x)+1$ for all $x \in I$.*

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 3.12 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].[11]

Lemma 3.12 ([HILL], cf. Appendix B) *If there exists an auxiliary-input false entropy generator on I, then there exists an auxiliary-input one-way function on I.*

Håstad et al. [HILL] actually show how to construct pseudorandom generators, rather than just one-way functions, but we only need a one-way function to establish the SZK/OWF Condition. This allows some steps in the construction to be omitted; see Appendix B for a proof of Lemma 3.12 (and a generalization, which we use to handle expected polynomial-time simulators in Section 7.3).

Proof of Lemma 3.10: 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 $\mathrm{H}(X'|Y') \geq r+1$ for some (X', Y') indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r-1$. (We may assume a gap of 2 rather than 1 by taking multiple independent samples from the joint distribution.)

[11] 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.

Let I be the set of instances $x \in \Pi_Y$ such that $\mathrm{H}(X|Y) < r$. First we show that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in **SZK**. We prove this by reducing Π' to Conditional Entropy Approximation. Indeed, the reduction is simply $x \mapsto ((X,Y), r+1)$. Then $\mathrm{H}(X|Y) \geq r$ when $x \in \Pi_Y \setminus I$ and $\mathrm{H}(X|Y) \leq r-1$ when $x \in \Pi_N$, as needed.

Now we show that we can construct a one-way function from instances $x \in I$. Intuitively, the fact that $\mathrm{H}(X'|Y') \geq r+1$ and (X',Y') is indistinguishable from (X,Y) means we should be able to extract $r+1$ *pseudorandom* bits from X given Y. That is, if we let H be a random hash function mapping to $r+1$ bits, then the distribution $(H, Y, H(X))$ is computationally indistinguishable from $(H, Y', H(X'))$, which we might hope to be statistically close to (H, Y', U_{r+1}) (because $\mathrm{H}(X'|Y') \geq r+1$), which in turn is computationally indistinguishable from (H, Y, U_{r+1}). However, the entropy of $(H, Y, H(X))$ equals $\mathrm{H}(H) + \mathrm{H}(Y) + \mathrm{H}(X|Y) < \mathrm{H}(H) + \mathrm{H}(Y) + r$, which is 1 bit less than the entropy of (H, Y, U_{r+1}). So, if this argument worked, then $(H, Y, H(X))$ would be a false entropy generator.

However, there are two (standard) difficulties in implementing this intuition. First, entropy (much less conditional entropy) is not a strong enough measure of randomness to allow extracting almost-uniform bits. (That is, it is not guaranteed that $(H, Y', H(X'))$ is statistically close to (H, Y', U_{r+1}).) Instead, we need a lower bound on (conditional) *min-entropy*, as required in the Leftover Hash Lemma (Lemma 2.7). Second, randomness extraction (e.g. as provided by the Leftover Hash Lemma) does not extract all the bits of min-entropy, but rather suffers an entropy loss related to the distance ε desired from uniform in the extracted bits. So we need a larger gap than one bit of entropy to tolerate this loss and still obtain a false entropy generator. Both of these difficulties are solved by taking direct products, *i.e.,* many independent samples of (X,Y). Taking a direct product has the effect of both (linearly) growing the entropy gap and converting entropy to min-entropy (with a sublinearly loss in entropy, as shown by the Flattening Lemma, Lemma 2.6).

We now proceed with the formal details. 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

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

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

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

which has entropy $s + k \cdot \mathrm{H}(Y) + (kr+1)$, which in turn is one bit larger than the entropy of D. Thus, we have constructed an auxiliary-input false entropy generator on I, and thus by Lemma 3.12 there exists a one-way function on I, as desired.

We now proceed to show that when $x \in \Pi_Y$, D is computationally indistinguishable from F. We know that there exist (X',Y') indistinguishable from (X,Y) such that $\mathrm{H}(X'|Y') \geq r+1$. By Lemma 2.2, we can 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 $(x,y) \in \mathrm{Supp}(X^*,Y^*)$.

By a hybrid argument (Lemma 2.13), D and F are computationally indistinguishable from

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

respectively, where the (X_i^*, Y_i^*)'s are independent copies of (X^*, Y^*).

Now we proceed to show that D^* is statistically indistinguishable from F^*, which will complete the proof. By Lemma 2.6, $\overline{X^*} = (X_1^*, \ldots, X_k^*)$ is Φ-flat given $\overline{Y^*} = (Y_1^*, \ldots, Y_k^*)$ for $\Phi = \sqrt{k} \cdot (m+n)$. 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

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

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

Thus, D^* is statistically close to the distribution $(H, \overline{Y^*}, H(W))$, which is 2^{-n}-close to $(H, \overline{Y^*}, U_{kr+1}) = F^*$ by the Leftover Hash Lemma (Lemma 2.7). This completes the proof. ■

3.4 The Indistinguishability Condition

In this section, we show that the Indistinguishability Condition is equivalent to the Conditional Pseudoentropy Condition, 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 complete problems for **SZK**. We note that the results of this section are not used later in the paper, except of course to establish the Indistinguishability Characterization of **ZK** (Theorem 3.4); they are included because this characterization may be of independent interest and may be of use in further studies of **ZK**.

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

Proof: The reduction is identical to the one used in the proof of Lemma 3.10 to construct a pseudoentropy generator on the instances in I. Let Π be a promise problem satisfying the Conditional Pseudoentropy Condition. As in the proof of Lemma 3.10, 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 $\mathrm{H}(X'|Y') \geq r + 1$ for some (X', Y') indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 1$. From X and Y, we can construct the samplable distributions D and F as in the proof of Lemma 3.10. In that proof, it is shown that when $x \in \Pi_Y$, then D and F are computationally indistinguishable. It is also shown that when $\mathrm{H}(X|Y) < r$ (in particular if $x \in \Pi_N$), then $\mathrm{H}(F) \geq \mathrm{H}(D) + 1$. By Lemma 2.1, this implies that $\Delta(D, F) \geq 1/2\ell$, where $\ell = \mathrm{poly}(n)$ is the number of bits output by D and F. Applying Lemma 2.3 and Lemma 2.13, 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 Condition. ■

Lemma 3.14 *If a promise problem satisfies the* Indistinguishability Condition*, then it satisfies the* Conditional Pseudoentropy Condition*.*

Proof: This is proven in the same way that we reduced Statistical Difference to Conditional Entropy Approximation in the proof of Proposition 3.2. Let Π be a promise problem satisfying the Indistinguishability Condition. This means that 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 $\Delta(X_0, X_1) \geq 2/3$ if $x \in \Pi_N$. 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 $x \in \Pi_Y$, the distributions 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 $x \in \Pi_N$, it holds that $\Delta(X_0, X_1) \geq 2/3$. Then, as in the proof of Proposition 3.2, we have $\mathrm{H}(B|Y) < .651 < 2/3$.

Thus the mapping $x \mapsto (B, Y), r = 1$ meets the requirements of the Conditional Pseudoentropy Condition, except that the gap in conditional entropies between the two cases is only $1 - 2/3 = 1/3$ bits. The gap can be amplified to 1 bit by taking direct products as usual. ■

4 From the SZK/OWF Condition to ZK

In this section, we construct a computational zero-knowledge proof system for every problem Π in **IP** that satisfies the SZK/OWF Condition. 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 construct a certain type of bit-commitment scheme in each of the two cases. The advantage is that the commitment schemes are easy to combine (via simple secret-sharing). We then use the combined commitment scheme in the generic compiler from **IP** to **ZK** [IY, BGG+].

4.1 Instance-Dependent Commitments

Recall that a *commitment scheme* is a two-phase protocol between a sender and a receiver. In the first phase, called the 'commit phase', the sender 'commits' to a private bit b. In the second phase, called the 'reveal phase', the sender reveals b and 'proves' that it was the bit to which she committed in the first phase. We require two properties of commitment schemes. The *hiding* property says that the receiver learns nothing about m in the first phase. The *binding* property says that after the commit phase, the sender is bound to a particular value of m; that is, she cannot successfully open the commitment to two different messages in the reveal phase. It is impossible to have commitment schemes that are both statistically hiding and statistically binding, but it is known how to construct commitment schemes that are computationally hiding and statistically binding assuming one-way functions exist [Nao, HILL]. In fact, this is the only way that one-way functions are used in the construction of computational zero-knowledge proofs for all of **IP** [GMW, IY, BGG+], and all the resulting theorems about **ZK** that rely on the assumption that one-way functions exist.

In this section, we will show how to use the fact that a promise problem Π satisfies the SZK/OWF Condition to construct a relaxed form of commitment scheme, tailored to Π, that still suffices for obtaining a zero-knowledge proof for Π. Specifically, we will construct an *instance-dependent commitment scheme* for Π. This 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 the 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. 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).

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 $G_0 \not\cong G_1$, then the commitment is perfectly binding. This idea was abstracted by Itoh, Ohta, and Shizuya [IOS], who studied the general utility of instance-dependent commitment schemes for constructing zero-knowledge proofs. Specifically, they showed that every language possessing a noninteractive instance-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. First, we allow the hiding property to be computational, since will use them to construct computational zero-knowledge proofs. Second, we only require security for an honest receiver (*i.e.,* one that follows the specified protocol); this means that the zero-knowledge proofs we construct with them will only be *honest-verifier* zero knowledge. However, since our honest-verifier zero-knowledge proofs will also be public coin (due to the instance-dependent commitments being public coin), we will be able to make them robust against cheating verifiers using the compiler of [GSV1]. Third, and most significantly, we allow the sender's algorithm to be computationally unbounded. This is okay when we use the instance-dependent commitments to construct zero-knowledge proofs, because the sender's role is played by the prover, who is allowed to be computationally unbounded. (Though this naturally renders the commitments useless for the application in [MV], which focused on prover efficiency.)

The fact that the sender is not polynomial time, however, 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, with a standard commitment scheme, one can assume without loss of generality that we have a 'canonical reveal phase', whereby the sender gives the message m and her coin tosses r to the receiver and the receiver checks that the transcript of the commit phase is consistent with m and r. (See [Gol4, Sec. 4.4.1].) This is not possible when the sender is computationally unbounded, because the receiver cannot run the sender's algorithm to check the transcript. Another example is that standard commitments are automatically "zero knowledge" in the sense that the receiver learns nothing (from both phases) other than the bit b 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 following definition.

Definition 4.1 *An (unbounded-sender, honest-receiver)* instance-dependent commitment scheme *for a promise problem* Π *consists of two interactive protocols* (S_1, R_1) *(the* commitment phase*) and* (S_2, R_2) *(the* reveal phase*) and a promise problem* $\mathrm{VAL} = (\mathrm{VAL}_Y, \mathrm{VAL}_N)$ *(capturing the 'valid'* *commitments). 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 allow S_1 and S_2 (resp., R_1 and R_2) to share the same coin tosses (as a way to maintain private state beyond the public commitment value z). 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). (In contrast, 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 $(x, z, b) \in \textsc{Val}_Y$ with probability $1 - \mathrm{neg}(n)$.*

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

4. *(Statistical 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* $\mathrm{neg}(n)$ *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 $1 - \mathrm{neg}(n)$, either $(x, z, 0)$ or $(x, z, 1)$ is in $\textsc{Val}_N$.*

The commitment scheme is called public coin *if it is public coin for the receiver R.*

A few remarks on the above conditions:

- As mentioned earlier, 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, without loss of generality, consist of S just sending its coin tosses to R.

- 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 instance-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 $\textsc{Val}$ (which has only two rounds). Both will be useful for us.

- Both the zero knowledge and hiding conditions are only required for honest 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] and the fact that our commitment schemes are public coin.

As shown in Section 6, our results yield characterizations of **ZK** and **SZK** in terms of instance-dependent commitment schemes:

Theorem 4.2 (Commitment Characterization of ZK)
$\Pi \in \mathbf{ZK}$ *if and only if* $\Pi \in \mathbf{IP}$ *and* Π *has a public-coin,* computationally *hiding instance-dependent commitment scheme in the sense of Definition 4.1.*

Theorem 4.3 (Commitment Characterization of SZK)
$\Pi \in \mathbf{SZK}$ *if and only if* $\Pi \in \mathbf{IP}$ *and* Π *has a* statistically *hiding instance-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. This intuition has been held by researchers for a number of years, based first on the construction of zero-knowledge proofs for all of **NP** and **IP** from commitment schemes [GMW, IY, BGG$^+$]. Partial converses were given by Damgård [Dam1, Dam2], who showed that every problem having a *3-round, public-coin* zero-knowledge proof has an instance-dependent commitment scheme[12] (as above, the commitment is statistically hiding if the proof system is statistical zero knowledge), and Ostrovsky and Wigderson [Ost, OW], who showed that zero-knowledge proofs for *hard-on-average* languages imply one-way functions (and hence standard commitment schemes [Nao, HILL]). As far as we know, the above theorems are the first to establish a genuine *equivalence* between zero-knowledge proofs and some form of commitment schemes.

In this section, we focus on proving the forward direction of Theorem 4.2:

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

We will prove this lemma by dealing separately with the **SZK** instances and OWF instances. This is done by combining two instance-dependent commitment schemes, one that is hiding on the "OWF instances" and the other that is hiding on the "**SZK** (yes) instances". For the OWF instances, we use a straightforward application of the known construction of commitment schemes from one-way functions [Nao, HILL].

Lemma 4.5 *If there exists an auxiliary-input one-way function on set* I*, then there is an instance-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 2.15, we can construct an auxiliary-input pseudorandom generator $\{G_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{3p(|x|)}\}$ on I. Now we adapt Naor's commitment scheme from pseudorandom generators [Nao]:

[12] Damgård's result is not stated in the language of instance-dependent commitments, but this formulation seems to follow from his technique.

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. Both parties set $v_1 = v$ and $v_0 = 0^{3p(n)}$.
2. S_1 chooses $r \leftarrow \{0,1\}^{p(n)}$ and sends $w = G_x(r) \oplus v_b$ to R_1.
3. The commitment z is defined to be the pair (v, w).

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

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

where again we define $v_1 = v$ and $v_0 = 0^{3p(|x|)}$. Clearly $\textsc{Val} \in \mathbf{NP}$, and in fact the reveal phase (S_2, R_2) simply consists of the sender S_2 providing the standard **NP** proof that $(x,(v,w),b) \in \textsc{Val}_Y$ (namely r such that $w = G_x(r) \oplus v_b$). Thus we have the required validity tests.

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 denote the message of R_1, we see that R_1's view of a commitment to 1, $(V, G_x(U_{p(n)}) \oplus V)$, is indistinguishable from $(V, U_{3p(n)} \oplus V) \equiv (V, U_{3p(n)})$, which in turn is indistinguishable from R_1's view of a commitment to 0, $(V, G_x(U_{p(n)}))$. 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 $G_x \oplus v$, 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 an instance-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 to prove Lemma 4.4.

Lemma 4.7 *If promise problems $\Pi = (\Pi_Y, \Pi_N)$ and $\Gamma = (\Gamma_Y, \Gamma_N)$ each have instance-dependent commitment schemes, then the promise problem $\Pi \cup \Gamma \overset{\text{def}}{=} (\Pi_Y \cup \Gamma_Y, \Pi_N \cap \Gamma_N)$ has an instance-dependent commitment scheme. If the commitment schemes for Π and Γ are both public coin, then so is the commitment scheme for $\Pi \cup \Gamma$. Moreover, the strategy of the sender in the commitment scheme for $\Pi \cup \Gamma$ on auxiliary input x can be implemented in probabilistic polynomial time given oracle access to the strategies of the senders in the commitment schemes for Π and Γ on auxiliary input x.*

Proof: Let (S', R') be the instance-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. The natural thing to do is for the sender to commit to two "shares" of its bit b, one with each scheme, and this is indeed what we do.

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

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

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

Reveal Phase $(S_2, R_2)(x, (z', z''), b)$**:** 1. S_2 sends b', b''.

2. R_2 checks that $b' \oplus b'' = b$ and rejects immediately if not.
3. S_2 and R_2 execute $(S_2', R_2')(x, z', b')$ and $(S_2'', R_2'')(x, z'', b'')$, 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 with high probability $(x, z', b') \in \textsc{Val}'_Y$ and $(x, z'', b'') \in \textsc{Val}''_Y$, and hence $(x, (z', z''), b) \in \textsc{Val}_Y$. (Here it is important that we required completeness to hold on all instances, rather than just on YES instances, since Π_Y and Γ_Y need not be the same.) The fact that (S_2, R_2) is an interactive proof for $\textsc{Val}$ follows by inspection, and the fact that $\textsc{Val} \in \mathbf{AM}$ follows from the fact that both $\textsc{Val}'$ and $\textsc{Val}''$ are in $\mathbf{AM}$ (combining the $\mathbf{AM}$ proof systems in the same way that we combined (S_2', R_2') and (S_2'', R_2'') to get (S_2, R_2)). For the zero-knowledge property, we have the new simulator $M(x, b)$ choose $b', b'' \leftarrow \{0,1\}$ such that $b' \oplus b'' = b$, run the original simulators $M'(x, b')$ and $M''(x, b'')$, and combine their outputs to simulate the view of R.

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 $(x, z', b') \notin \textsc{Val}'_N$ and there exists at most one $b'' \in \{0,1\}$ such that $(x, z'', b'') \notin \textsc{Val}''_N$. Thus there is at most one b (namely $b = b' \oplus b''$) such that $(x, (z', z''), b) \notin \textsc{Val}_N$, as desired.

By inspection, the above transformation maintains public coins and the sender's complexity. ■

Putting the above together, we can prove Lemma 4.4, stating that every language satisfying the SZK/OWF CONDITION has an instance-dependent commitment scheme.

Proof Lemma 4.4: Let Π be any promise problem satisfying the SZK/OWF CONDITION, and let $I \subseteq \Pi_Y$ be the set of "OWF instances". Since $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in $\mathbf{SZK}$, Lemma 4.6 gives us an instance-dependent commitment scheme for Π', with public coins and a sender that is PPT given an **NP** oracle. Since we have an auxiliary-input one-way function on I, Lemma 4.5 gives us an instance-dependent commitment scheme for $\Gamma = (I, \overline{I})$, with public coins and a PPT sender. Combining these via Lemma 4.7, we get an instance-dependent commitment scheme for $\Pi' \cup \Gamma = \Pi$, with public coins a sender that is PPT given an **NP** oracle. ■

4.2 The Zero-Knowledge Proof

We now show that to obtain a zero-knowledge proof for a problem $\Pi \in \mathbf{IP}$, it suffices for Π to have an instance-dependent commitment scheme in the sense of the previous section. This is done by simply using the instance-dependent commitment scheme to implement the **IP**-to-**ZK** compiler of [GMW, IY, BGG$^+$].

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

Proof: We begin with the special case that $\Pi \in \mathbf{NP}$, where we follow the approach of Itoh, Ohta, and Shizuya [IOS] using our more general notion of instance-dependent commitments. The idea is to use the zero-knowledge proofs of Goldreich, Micali, and Wigderson [GMW] for all of **NP**, replacing the commitment scheme used there with the instance-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 an arbitrary 3-coloring C_0 of G, and lets C be the coloring obtained by permuting the three colors in C_0 uniformly at random.
3. P commits to the color of each vertex under C by engaging with V in (polynomially many executions of) the commitment phase of instance-dependent commitment scheme for Π on common input x.
4. V selects a random edge e in G and sends e to P.
5. P reveals the colors of the endpoints of e, and proves their validity to V via the reveal phase of the instance-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 follows from completeness of the instance-dependent commitment scheme. Soundness follows from the binding property of the instance-dependent commitment scheme when $x \in \Pi_N$.

(Honest-verifier) zero knowledge follows from the hiding and zero-knowledge properties of the commitment scheme when $x \in \Pi_Y$. 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 assigns arbitrary colors for the rest of the 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. (Note that here, unlike in [GMW], we deal with an honest verifier and thus the verifier's challenge e is equivalent to its coin tosses and is indeed chosen uniformly at random.) The computational (resp., statistical) hiding property of the commitment scheme implies that this simulation is computationally (resp., statistically) 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 instance-dependent commitment scheme on common input x. (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 chooses and 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 the zero knowledge protocol described above (*i.e.,* GMW with instance-dependent commitments): there exist values $b_1, \ldots, b_m$ 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. ■

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 zero-knowledge proof system can be transformed into a (cheating-verifier) public-coin zero-knowledge proof system. Furthermore,*

1. *The resulting proof system has twice as many rounds as the original one.*

2. *The resulting prover strategy on any input x can be implemented in probabilistic polynomial time given oracle access to the original prover strategy on the same input x.*

3. *The resulting proof system has completeness error $2^{-\Omega(n)}$ and soundness error $1/n$ on input length n. In case the original proof system has perfect completeness, so does the resulting one.*

4. *If the original proof system is statistical zero knowledge, then so is the resulting proof system.*

5. *The resulting proof system has a black-box simulator.*

Note that the above theorem provides a zero-knowledge proof with nonnegligible soundness error (namely $1/n$). This can be reduced to negligible error by performing $\omega(1)$ sequential repetitions.

5 Instance-dependent commitments for SZK

In this section, we construct our instance-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 an instance-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, without loss of generality, 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 x, and the receiver verifies that $x = X_b(r)$.

When X_0 and X_1 are statistically close, this scheme is indeed hiding. On the other hand, 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 binding. 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 get around this difficulty, we notice that the intersection of 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 $\mathrm{H}(X_0) = \mathrm{H}(X_1) = h$). Note that there can be many ($\gg 2^h$) such elements. Second, there can be samples that are not atypically light for either distribution. However, it can be shown that there can only be a relatively few ($\ll 2^h$) elements of this second type, provided the distributions have statistical difference exponentially close to 1. Still, we need to cope with both kinds of samples.

To deal with these problems, we replace both the commit phase and reveal phase with interactive protocols. The commit phase protocol 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. The fact that this is feasible relies on the fact that there are relatively few samples that are not atypically light for both distributions. The reveal phase protocol, then, is simply an interactive proof that the sample is not atypically light for X_b.

Needless to say, 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 well-suited for these tasks. Specifically, 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 these protocols is that the sender can no longer be implemented in probabilistic polynomial time (but rather in $\mathbf{BPP}^{\mathbf{NP}}$), and also that the round complexity becomes polynomial rather than being a 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 from 1/3 and 2/3 to be 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:*

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

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 $\Pi \in$ **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 $\Delta(Z_0, Z_1) \le \ell \cdot 2^{-k}$.*
- *If $x \in \Pi_N$, then $\Delta(Z_0, Z_1) \ge 1 - 2^{-\ell}$.*
- *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.*

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. Specifically, we can take k to be linear in n and ℓ to be a large polynomial in n and have the deviation from flatness $\sqrt{\ell} \cdot \mathrm{poly}(n,k)$ remain sublinear in ℓ. We will show (in Lemma 5.3 below) 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 $\Pi \in$ **SZK** and the parameters 1^k and 1^ℓ 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

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

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

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

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 2.3,

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

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 (Lemma 2.6) 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 relatively small number of strings that are non-light 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 $\Phi > 0$,*

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

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

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

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

Note that the above lemma gives us a useful bound ($\ll 2^{\mathrm{H}(Z_b)}$) when the slackness parameter of Φ is smaller than ℓ. Fortunately, Lemma 5.2 allows us to obtain $\Phi = o(\ell)$ while still having a statistical difference of $1 - 2^{-\ell}$ on NO instances.

5.3 Okamoto's protocols

We now describe the two protocols of Okamoto [Oka] 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 any 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].

Below, all distributions are given in the form of circuits that generate them. The input to these protocols will include 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.

Definition 5.4 (sample generation protocol) *A protocol (S, R) is called a* sample generation protocol *if on common input a distribution X, specified by a circuit with m input gates and n output gates, and parameters Φ and t, such that X is Φ-flat and $1 \leq t \leq \Phi$, the protocol yields a common output in $\{0,1\}^n$ such that the following holds:*

1. *(Efficiency) R is computable in probabilistic polynomial time.*
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 $2\sqrt{t\Phi} \cdot \Phi$-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 $2^{-6\sqrt{t\Phi}\cdot\Phi} \cdot 2^{\mathrm{H}(X)}$, 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, Φ, 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, Φ, t) and output the view of R, appended by the output.*

   (B) *Choose $x \leftarrow X$ and output $(M(X, \Phi, t, x), x)$.*

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. (Our proof that the protocol satisfies the second soundness condition 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 random view consistent with 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'). 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 sample generation protocol. Furthermore, the protocol is public coin, the sender strategy is computable in probabilistic polynomial time with an* **NP** *oracle, and the number of messages exchanged in the protocol is linear in m, the input length of the sampling circuit for the input distribution X.*

Actually, in [Oka, GV], the sample generation protocol is not shown to satisfy the Soundness II condition of Definition 5.4 nor the bound on sender complexity specified in Theorem 5.5. Thus we repeat the description of the protocol here.

Sample generation protocol (S, R)**:**
Input: (X, Φ, t), where X has m input gates and n output gates and $t \leq \Phi$

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 $\{0,1\}^{m-3t\Phi}$ and send h_i to S.
   (b) S: Choose (r_{i-1}, x_i) from the distribution $\{r : X(r) = x_{i-1}\} \otimes X$, 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}.

The sender complexity claimed in Theorem 5.5 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 5.6 *The above protocol satisfies the Soundness II condition of Definition 5.4.*

Proof: Fix a set T of size at most $2^{-6\sqrt{t\Phi}\cdot\Phi} \cdot 2^{\mathrm{H}(X)}$. 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 $2\sqrt{t\Phi}\cdot\Phi$-heavy with probability at most $m \cdot 2^{-\Omega(t^2)}$. In fact, the proof of this condition in [GV] also shows that x_{m-1} is $2\sqrt{t\Phi}\cdot\Phi$-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 $2\sqrt{t\Phi}\cdot\Phi$-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

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

where the inequality is due to x_{m-1} not being $2\sqrt{t\Phi}\cdot\Phi$-heavy. 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

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

where the last inequality uses $t \leq \Phi$. Since $h_m(z)$ is uniformly distributed in its range $\{0,1\}^{m-3t\Phi}$ 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 5.7 (sample test protocol) *A protocol (S, R) is called a* sample test protocol *if on common input a distribution X, specified by a circuit with m input gates and n output gates, a string $x \in \{0,1\}^n$ and parameters Φ, t, such that X is Φ-flat and $t \le \Phi$, the following holds:*

1. *(Efficiency) R is computable in probabilistic polynomial time.*
2. *("Completeness") If both parties are honest and x is $t{\cdot}\Phi$-typical then R accepts with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$.*
3. *("Soundness") If x is $6\sqrt{t\Phi} \cdot \Phi$-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, Φ, t) as above and for every $t \cdot \Phi$-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, Φ, t) and output the view of R.*

   (B) *Choose r uniformly in $\{r' : X(r') = x\}$, and output $M(X, x, \Phi, t, r)$.*

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 5.8 (implicit in [Oka], explicit in [GV]) *There exists a public-coin sample test protocol. Furthermore, the protocol is public coin, the sender strategy is computable in probabilistic polynomial time with an* **NP** *oracle, and 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 instance-dependent commitments for all of **SZK**, and thereby prove Lemma 4.6. 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 to be determined later. Applying the reduction of Lemma 5.2 to x, we obtain distributions (Z_0, Z_1) such that

- If $x \in \Pi_Y$, then $\Delta(Z_0, Z_1) \le \ell \cdot 2^{-k} < 2^{-n}$.
- If $x \in \Pi_N$, then $\Delta(Z_0, Z_1) \ge 1 - 2^{-\ell}$.
- For all x, it holds that $\mathrm{H}(Z_0) = \mathrm{H}(Z_1)$ and both Z_0 and Z_1 are Φ-flat for $\Phi = \sqrt{\ell}{\cdot}\mathrm{poly}(n, k) < n^{4c}$, when c is sufficiently large.

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 $\mathrm{H}(Z_0) \le \mathrm{H}(Z) \le \mathrm{H}(Z_0) + 1$. We also claim that Z inherits the flatness of Z_0 and Z_1.

Claim 5.9 *Z is 3Φ-flat.*

The tedious proof of this claim is deferred to Appendix A. Now we construct the instance-dependent commitment scheme (S, R) as follows, setting $t = n$:

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

1. S_1 and R_1 execute the sample generation protocol of Theorem 5.5 on input $(Z, 3\Phi, t)$ to obtain output z, where Z, Φ, and t are as defined above.
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

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

Reveal Phase $(S_2, R_2)(x, (z, d), b)$**:**
S_2 and R_2 execute the sample test protocol of Theorem 5.8 on the input $(Z_{d\oplus b}, z, \Phi, t)$, and R_2 accepts or rejects according to its outcome.

Claim 5.10 *The above protocol is a statistically hiding instance-dependent commitment scheme in the sense of Definition 4.1.*

Proof:

1. (Receiver's 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, where $m = \text{poly}(n)$ is the number of input gates of the circuit generating Z. Thus (c, r) has statistical difference at most 2^{-n} from uniform. If (c, r) were uniformly distributed, then by the Φ-flatness of $Z_{c\oplus d}$, the probability (over r) that $z = Z_c(r)$ is $t\Phi$-typical for $Z_c = Z_{d\oplus b}$ is least $1 - 2^{-t^2} > 1 - 2^{-n}$. Therefore, $(x, (z, d), b) \in \textsc{Val}_Y$ 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 show that $\textsc{Val}$ is in **AM**, we design an **AM** proof system for it as follows. On input $(x, (z, d), b)$, the prover sends an approximation k to $\mathrm{H}(Z_{d\oplus b})$, and then proves that (a) $\mathrm{H}(Z_{d\oplus b}) \lesssim k$, and (b) $|\{r : Z_{d\oplus b}(r) = z\}| \gtrsim 2^m \cdot 2^{-k-t\Phi}$, where again m is number of input gates of the circuit generating Z. Step (a) can be done because approximating entropy to within an additive constant (say ±1) is in

AM $\cap$ **co-AM** [PT].[13] Step (b) can be done using an **AM** protocol for proving approximate lower bounds on the sizes of efficiently recognizable sets [Sip, Sto, BM]. Proving (a) and (b) suffices because on YES instances of VAL, we have

$$|\{r : Z_{d\oplus b}(r) = z\}| \geq 2^m \cdot 2^{-\mathrm{H}(Z_{d\oplus b})-t\Phi},$$

and on NO instances we have

$$|\{r : Z_{d\oplus b}(r) = z\}| \leq 2^m \cdot 2^{-\mathrm{H}(Z_{d\oplus b})-6\sqrt{t\Phi}\cdot\Phi} < 2^m \cdot 2^{-\mathrm{H}(Z_{d\oplus b})-t\Phi-5}.$$

4. (Zero Knowledge) This 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 $(Z, 3\Phi, t, z)$ to obtain a transcript γ_1, runs the simulator for the sample test protocol on $(Z_c, z, \Phi, t, (c, r))$ 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 according to the conditional distribution $C|_{Z_C=z}$, where C is uniform in $\{0, 1\}$. We have seen above that the sample generation protocol generates z according to a distribution that has statistical difference at most 2^{-n} from a random sample of $Z \equiv Z_C$. Thus the pair (z, c) is generated according to a distribution having statistical difference at most 2^{-n} from (Z_C, C). In the case of a YES instance, where Z_0 and Z_1 have statistical difference at most 2^{-n}, (Z_C, C) has statistical difference at most 2^{-n} from (Z_C, C') where C' is a random bit independent of C. Thus R_1's view in case $b = 0$ is statistically indistinguishable from R_1's view in case $b = 1$.

6. (Statistically binding on NO instances) Let $T = \{z : z \text{ is not } 6\sqrt{t\Phi}\cdot\Phi\text{-light for } Z_0 \text{ nor for } Z_1\}$. By Lemma 5.3,
$$|T| \leq \frac{2^{\mathrm{H}(Z_0)}}{2^{\ell-6\sqrt{t\Phi}\cdot\Phi}} \leq 2^{-6\sqrt{t\Phi}\cdot\Phi} \cdot 2^{\mathrm{H}(Z)},$$
where the last inequality is because $\ell = n^{7c} > 12n^{6c+.5} > 12\sqrt{t\Phi}\cdot\Phi$. 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 $6\sqrt{t\Phi}\cdot\Phi$-light for $Z_{d\oplus b}$. That is, there is at most one value of b such that $(x, (z, d), b) \notin \mathrm{VAL}_N$, as desired.

■

Proof of Lemma 4.6. Using Claim 5.10, all that is left is to verify is that the protocol is public coin and the sender can be implemented in probabilistic polynomial time given an **NP** oracle. Both of these follow from the analogous properties of the Sample Generation and Sample Test protocols given in Theorems 5.5 and 5.8. ■

[13] Indeed, the promise problem ENTROPY APPROXIMATION (EA), where $\mathrm{EA}_Y = \{(X, k) : \mathrm{H}(X) \geq k + 1\}$, $\mathrm{EA}_N = \{(X, k) : \mathrm{H}(X) \leq k\}$ is complete for noninteractive statistical zero knowledge (**NISZK**) [GSV2], and **NISZK** $\subseteq$ **SZK** $\subseteq$ **AM** $\cap$ **co-AM** [For, AH].

6 Putting it Together

Now we can put together the results proven in the previous three sections and establish Theorems 1.2, 3.4, 3.6, and 4.2.

Theorem 6.1 (ZK Characterization Theorem) *For a promise problem* Π*, the following conditions are equivalent:*

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

Proof:

1 $\Rightarrow$ **2** This follows from Lemma 3.7, together with the trivial inclusion $\mathbf{HVZK} \subseteq \mathbf{IP}$.

2 $\Rightarrow$ **3** This is Lemma 3.10.

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

5 $\Rightarrow$ **7** Suppose $\Pi \in \mathbf{IP}$ and Π has a public-coin instance-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 $\Rightarrow$ **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 beyond $\mathbf{BPP}^{\mathbf{NP}}$). 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 instance-dependent commitment can be implemented in probabilistic polynomial time with an **NP** oracle. Same for Theorem 4.9.

7/8 $\Rightarrow$ **6** $\Rightarrow$ **1** These are immediate from the definitions.

2 $\Leftrightarrow$ **4** This is by Lemmas 3.13 and 3.14.

■

We also prove Theorem 4.3, which we restate here:

Theorem 6.2 (Thm. 4.3, restated) $\Pi \in \mathbf{SZK}$ *if and only if* $\Pi \in \mathbf{IP}$ *and* Π *has a* statistically *hiding instance-dependent commitment scheme in the sense of Definition 4.1.*

Proof:

$\Rightarrow$ This follows from Lemma 4.6, together with the trivial inclusion $\mathbf{SZK} \subseteq \mathbf{IP}$.

$\Leftarrow$ This follows from Lemma 4.8, together with the fact that $\mathbf{HVSZK} = \mathbf{SZK}$ [Oka, GSV1].

■

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 Theorem, 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. Specifically, we use the SZK/OWF Characterization Theorem to deduce the Ostrovsky–Wigderson theorems about **ZK** from the earlier (and much simpler) work of Ostrovsky [Ost] on **SZK**. In fact, we only need our results from Section 3, showing that every problem in **HVZK** satisfies the SZK/OWF Condition. Our results in the converse direction, from Sections 4, 5, and 6, are not needed.

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 every polynomial $r(n)$, there exists an infinite set $I \subseteq \{0,1\}^*$ such that*

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

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 $\mathbf{HVZK} \not\subseteq$ **P/poly**.

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

Definition 7.2 *A promise problem* Π *is* hard on average *if there exists a probabilistic polynomial-time sampling algorithm* S*, a polynomial* r*, and a constant* $\delta > 0$ *such that for every nonuniform PPT* A*, the following holds for all but finitely many* n*:*

$$\Pr_{x \leftarrow S(1^n)}\left[(x \in \Pi_Y \cup \Pi_N) \wedge (A(x) \neq \chi_\Pi(x)) \wedge |x| \geq n^\delta\right] \geq \frac{1}{r(n)},$$

where χ_Π *is the characteristic function of* Π*, i.e.,* $\chi_\Pi(x) = 1$ *if* $x \in \Pi_Y$*,* $\chi_\Pi(x) = 0$ *if* $x \in \Pi_N$*, and* $\chi_\Pi(x) = \star$ *otherwise.*

Theorem 7.3 ([OW, Thm 2]) *If* **HVZK** *contains a hard-on-average promise problem, then (standard) one-way functions exist.*

We begin by observing that the SZK/OWF Characterization (Theorem 1.2) immediately implies a stronger form of one-way functions than given by Theorem 7.1 under the stronger (but still worst-case) hypothesis that **HVZK** $\neq$ **HVSZK**.

Theorem 7.4 *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*

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

for all sufficiently long $x \in I$*.*

The key difference between the conclusions of Theorem 7.1 and Theorem 7.4 is that the order of quantifiers between the adversary A and the infinite set I is reversed. In the former, the infinite set of 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.

Recall that **HVSZK** $\subseteq$ **AM** $\cap$ **co-AM** [For, AH], and thus it is unlikely that **NP** $\subseteq$ **HVSZK**. Thus Theorem 7.4 can be interpreted as further evidence, incomparable to what is given by the Ostrovsky–Wigderson Theorems (Thms. 7.1 and 7.3), that one-way functions are necessary to construct zero-knowledge proofs for all of **NP** (not to mention all of **IP**). (Recall, that it is known that one-way functions are *sufficient* to establish that **IP** = **ZK**. [GMW, IY, BGG$^+$, Nao, HILL].)

Proof of Theorem 7.4: Suppose **HVZK** $\neq$ **HVSZK**, and let Π be any promise problem in **HVZK****HVSZK**. By Theorem 6.1, Π satisfies the SZK/OWF Condition. 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 alternate proofs of the Ostrovsky–Wigderson theorems themselves based on the work of Ostrovsky on **SZK**, as captured in the following theorem:

Theorem 7.5 (implicit in Ostrovsky [Ost]) *For every problem* $\Pi \in \mathbf{HVSZK}$*, there exists a poly-time auxiliary-input function ensemble* $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}_{x \in \{0,1\}^*}$*, a probabilistic polynomial-time oracle machine* M*, and a negligible function* ϵ *such that for every* $x \in \Pi_Y \cup \Pi_N$*, every* $t \in \mathbb{N}$*, and every function* $A : \{0,1\}^{q(|x|} \to \{0,1\}^{p(|x|)}$*, we have*

$$\Pr\left[A(f_x(U_{p(|x|)})) \in f_x^{-1}(f_x(U_{p(|x|)}))\right] > \epsilon(|x|) + \frac{1}{t}$$
$$\Rightarrow \quad \Pr\left[M^A(x, 1^t) = \chi_\Pi(x)\right] \geq 1 - 2^{-|x|},$$

where χ_Π *is again the characteristic function of* Π.

Note that t, which specifies A's success probability in inverting f_x (upto a negligible term), is given as an input (in unary) to the oracle machine M. Intuitively, for M to take advantage of the fact that A inverts f_x with probability $\approx 1/t$, M must be allowed running time polynomially related to t.

Proof of Theorem 7.1: Suppose that $\mathbf{HVZK} \neq \mathbf{BPP}$. Then either $\mathbf{HVZK} \neq \mathbf{HVSZK}$ or $\mathbf{HVSZK} \neq \mathbf{BPP}$. In the first case, we are done by Theorem 7.4. Thus, we need only show that $\mathbf{HVSZK} \neq \mathbf{BPP}$ implies the existence of an auxiliary-input family of functions that is not easy to invert. This follows readily from Theorem 7.5. Let Π be any promise problem in $\mathbf{HVSZK} \setminus \mathbf{BPP}$, and let $\{f_x\}$ be the family of functions provided by Theorem 7.5. If there is a uniform PPT A inverting f_x with probability at least $1/r(|x|)$, for some polynomial r and all but finitely many x, then by Theorem 7.5, $M^{A(x,\cdot)}(x, 1^{2r(|x|)2})$ is PPT algorithm that decides Π correctly for all but finitely many x.[14] This contradicts the assumption that $\Pi \notin \mathbf{BPP}$. ■

The above proof illustrates why Theorem 7.1 only yields a family of functions that is not easy to invert, rather than the stronger notion of auxiliary-input one-way functions achieved in Theorem 7.4. The reason is that the supposed inverter A for the family of functions is used to construct a **BPP** algorithm for the promise problem $\Pi \in \mathbf{SZK}$. 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.

How is this difficulty avoided in Theorem 7.4, which relies on the SZK/OWF Condition as established in Section 3? Intuitively, the reason is that the hardness of inverting the function f_x of the SZK/OWF Condition when x is a "OWF instance" is not derived from the intractability of the promise problem Π, which does not make sense for fixed instances x (for the reasons discussed above), but rather is based on the intractability of distinguishing the output of the simulator from the the real interaction in an **HVZK** proof system (which makes sense for fixed instances x and indeed is required to hold for every $x \in \Pi_Y$).

[14] A minor technicality is that Theorem 7.5 is stated for deterministic oracles A, whereas here A may be probabilistic. However, after a standard error reduction by $O(r(|x|))$ repeated trials, we can ensure that with probability .99 over A's coin tosses w, the deterministic algorithm $A(x, \cdot; w)$ inverts $f_x(U_{p(|x|)})$ with probability $(3/4) \cdot (1/r(|x|))$. So we obtain a **BPP** algorithm for Π by randomly choosing w and running $M^{A(x,\cdot,w)}(x, 1^{2r(|x|)})$.

Theorem 7.3 gets around the difficulty in a different way, by requiring a stronger form of intractability for the problem Π, namely that it is hard on average. Let us first consider the case that we have a hard-on-average problem $\Pi \in \mathbf{HVSZK}$, following Ostrovsky [Ost]. Instead of hoping that x's membership in Π will be hard to decide and thus f_x from Theorem 7.5 hard to invert for particular values of x, we simply can sample a random instance x and be guaranteed, by the definition of "hard on average," that for *any* polynomial-time algorithm A, the instance x will be "hard" for A with at least a fixed nonnegligible probability. Thus $f(x, y) = (x, f_x(y))$ will be hard to invert with a fixed nonnegligible probability for any polynomial-time inverter. Now to handle the more general case of $\Pi \in \mathbf{HVZK}$, we use the SZK/OWF Condition, combining Ostrovsky's one-way functions just described, which are hard to invert in case x is an "SZK instance", with the one-way functions of the SZK/OWF Condition, which are hard to invert in case x is a "OWF instance." This yields the following new proof of Theorem 7.3.

Proof of Theorem 7.3: Suppose $\Pi \in \mathbf{HVZK}$ is hard-on-average with respect to the sampling algorithm S. Theorem 6.1 tells us that Π satisfies the SZK/OWF Condition, so there is a set $I \subseteq \Pi_Y$ and a poly-time auxiliary-input function ensemble $\mathcal{F} = \{f_x : \{0,1\}^{p(|x|)} \to \{0,1\}^{q(|x|)}\}$ such that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in $\mathbf{SZK} = \mathbf{HVSZK}$ and $\mathcal{F}$ is one-way on I. We now apply Theorem 7.5 to Π' to get another poly-time auxiliary-input function ensemble $\mathcal{F}' = \{f'_x : \{0,1\}^{p'(|x|)} \to \{0,1\}^{q'(|x|)}\}$ such that any inverter for f'_x can be used to decide whether x is a YES or NO instance of Π'.

Now we construct a one-way function g_n, where n is the security parameter, as follows: the input to g_n is a triple (r, w, w'). To compute $g_n(r, w, w')$, we interpret r as coin tosses for the sampling algorithm S, obtaining an instance $x = S(1^n; r)$ of Π, and output $(x, f_x(w), f'_x(w'))$.

We will now argue that g_n is a *weak* one-way function, namely that no nonuniform PPT algorithm can invert g_n with probability higher than $1 - 1/(4r(n))$, where r is the polynomial in the definition of hard on average. Suppose that there is a nonuniform PPT inverter A such that

$$\Pr\left[A(g_n(R, W, W')) \in g_n^{-1}(g_n(R, W, W'))\right] \geq 1 - 1/(4r(n))$$

for infinitely many n, when R, W, and W' are chosen uniformly at random from the bit-strings of appropriate length. (Since A is nonuniform, we may assume that it is deterministic without loss of generality.)

First, we note that when $X = S(1^n; R) \in I$, then A has only a negligible probability of inverting over the choice of W, by the one-wayness of f_X. Thus, we have

$$\Pr\left[A(g_n(R, W, W')) \in g_n^{-1}(g_n(R, W, W')) \wedge S(1^n; R) \notin I\right] \geq 1 - 1/(3r(n)). \tag{2}$$

Now we use Theorem 7.5 to convert A into an algorithm B that decides Π', and hence Π, well on average (with respect to the distribution $S(1^n)$). Specifically, on input x, B chooses w uniformly at random and runs $M^{A(x, f_x(w), \cdot)_3}(x, 1^{|x|})$, where δ is the constant in the definition of hard on average and $A(x, f_x(w), \cdot)_3$ denotes the third component of the output of A.

From Equation (2), it follows that with probability at least $1 - 2/(3r(n))$ over the choice of $r \leftarrow R$ and $w \leftarrow W$, we have $x = S(1^n; r) \notin I$ and

$$\Pr\left[A(x, f_x(w), f'_x(W'))_3 \in (f'_x)^{-1}(f'_x(W'))\right] \geq \Pr\left[A(g_n(r, w, W')) \in g_n^{-1}(g_n(r, w, W'))\right] \geq 1/2, \tag{3}$$

where the probabilities are taken only over W'. Whenever Inequality (3) holds and we have $x \in \Pi'_Y \cup \Pi'_N = (\Pi_Y \cup \Pi_N) \setminus I$, Theorem 7.5 ensures that $M^{A(x, f_x(w), \cdot)_3}(x, 1^{|x|})$ correctly decides whether

x is a YES or NO instance of Π' with probability at least $1 - 2^{-|x|}$. Thus, setting $X = S(1^n; R)$, we have

$$\Pr\left[(X \in \Pi_Y \cup \Pi_N) \wedge (B(X) \neq \chi_\Pi(X)) \wedge (|X| \geq n^\delta\right] \leq 2/(3r(n)) + 2^{-n^\delta} < 1/r(n).$$

This contradicts the fact that Π is hard on average with respect to the distribution $S(1^n)$. ■

We note that an alternative way prove a version of Theorem 7.3 is to combine our results with [SV, Thm 5.12], which shows that if a hard-on-average problem satisfies the INDISTINGUISHABILITY CONDITION, then one-way functions exist. However, [SV, Thm 5.12] uses a stronger definition of hard on average than Definition 7.2, requiring that any PPT has error probability negligibly close to 1/2, rather than just $1/\text{poly}(n)$. In addition, we feel that it is informative to see how the result for **ZK** follows from combining Ostrovsky's work on **SZK** (*i.e.,* Theorem 7.5) with the SZK/OWF CONDITION.

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 $\Pi, \Gamma \in$ **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 $\Pi \in$ **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. Such closure properties were previously known for **SZK** [DDPY, Oka, SV].[15] Indeed we prove our results by reduction to the **SZK** case via the SZK/OWF Characterization Theorem. (An alternative way of proving the results is to mimic the proofs for **SZK**, replacing STATISTICAL DIFFERENCE in the construction of [SV] with the INDISTINGUISHABILITY CONDITION.)

Theorem 7.6 **ZK** *is closed under union.*

Proof: By Theorem 1.2, a promise problem is in **ZK** if and only if it is in **IP** and it satisfies the SZK/OWF CONDITION. Since **IP** is closed under union, it suffices to show that the class of problems satisfying the SZK/OWF CONDITION is closed under union.

Suppose that Π and Γ satisfy the SZK/OWF CONDITION. Then there are sets I and J and poly-time auxiliary-input families of functions $\{f_x\}$, $\{g_x\}$ such that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ and $\Gamma' = (\Gamma_Y \setminus J, \Gamma_N)$ are both in **SZK**, f_x is one-way when $x \in I$ and g_x is one-way when $x \in J$. We claim that the set $K = I \cup J$ of "OWF instances" and the family of functions $\{h_x\}$ where $h_x(y, z) = (f_x(y), g_x(z))$ meet the requirements for showing that $\Pi \cup \Gamma$ satisfies the SZK/OWF CONDITION. Indeed, when $x \in K$, then h_x is one-way because either f_x or g_x is one-way. The promise problem $((\Pi_Y \cup \Gamma_Y) \setminus K, \Pi_N \cap \Gamma_N)$ is in **SZK** because it is a restriction of the promise problem $\Pi' \cup \Gamma' = (\Pi'_Y \cup \Gamma'_Y, \Pi'_N \cap \Gamma'_N)$ (*i.e.,* the YES instances of the former problem are a subset of those of the latter and the NO instances of both problems are the same), and $\Pi' \cup \Gamma'$ in **SZK** because **SZK** is closed under union [Oka]. ■

[15] In fact, since **SZK** is closed under complement [Oka], its closure properties extend even to non-monotone formulae.

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 7.7 *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 7.8 *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 7.9 *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 SZK/OWF Condition) 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-parameterized definition of zero knowledge (cf. Remark 5 at the end of Section 2.5).

Theorem 7.10 *For any promise problem $\Pi \in$ **SZK** and any constant $\delta > 0$, $\mathrm{Mon}_\delta(\Pi) \in$ **SZK**.*

Proof: First we note that **IP** is closed under $\mathrm{Mon}_\delta(\cdot)$: to prove that $(\phi, x_1, \ldots, x_k)$ is in $\mathrm{Mon}_\delta(\Pi)_Y$, it suffices to prove that a subset of the x_i's are in Π_Y, due to the monotonicity of ϕ. Thus, by Theorem 1.2 we need only show that if Π satisfies the SZK/OWF CONDITION, then $\mathrm{Mon}_\delta(\Pi)$ satisfies the SZK/OWF CONDITION.

Let Π be any promise problem satisfying the SZK/OWF CONDITION, with corresponding set $I \subseteq \Pi_Y$ and poly-time auxiliary-input functions $\{f_x\}$ such that $\Pi' = (\Pi_Y \setminus I, \Pi_N)$ is in **SZK** and f_x is hard to invert when $x \in I$. Since **SZK** is closed under $\mathrm{Mon}_\delta(\cdot)$ (even for $\delta = 0$) [DDPY, SV], we have that $\mathrm{Mon}_\delta(\Pi') \in$ **SZK**. Note that $\mathrm{Mon}_\delta(\Pi')$ is identical to $\mathrm{Mon}_\delta(\Pi)$ except on instances $(\phi, x_1, \ldots, x_k)$ where at least one x_i is in I, because then $\chi_\Pi(x_i) = 1$ but $\chi_{\Pi'}(x_i) = \star$. Specifically, since changing a variable's assignment from 1 to $\star$ can only change the value of a monotone formula from 1 to $\star$, we have $\mathrm{Mon}_\delta(\Pi')_N = \mathrm{Mon}_\delta(\Pi)_N$ and $\mathrm{Mon}_\delta(\Pi')_Y = \mathrm{Mon}_\delta(\Pi)_Y \setminus J$, where

$$J = \{(\phi, x_1, \ldots, x_k) \in \mathrm{Mon}_\delta(\Pi)_Y : \exists i \; x_i \in I\}.$$

Thus, to show that $\mathrm{Mon}_\delta(\Pi)$ satisfies the SZK/OWF CONDITION, it suffices to show that we can construct a one-way function from any instance in J. To do this, we simply define

$$g_{(\phi, x_1, \ldots, x_k)}(y_1, \ldots, y_k) = (f_{x_1}(y_1), \ldots, f_{x_k}(y_k)).$$

Then when $x = (\phi, x_1, \ldots, x_k) \in J$, there is at least one f_{x_i} that is hard to invert (by nonuniform PPT algorithms running in time $\mathrm{poly}(|x_i|) = \mathrm{poly}(|x|)$, since $|x| \geq |x_i| \geq |x|^\delta$) implying that g is hard to invert. ■

Theorem 7.10 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 7.11 (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 7.7). We further consider the case where we restrict the complexity of computing the output of the reduction from the queries:

Definition 7.12 ($\mathbf{NC}^1$ 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 7.10 as follows:

Corollary 7.13 **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. ■

Consequences for knowledge complexity. 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 7.14 (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 $k : \mathbb{N} \to \mathbb{N}$ *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. *$|h(x)| = k(|x|)$.*
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 7.13, we can prove a collapse in this hierarchy:

Theorem 7.15 *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 [Gol4], 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 original Goldwasser–Micali–Rackoff [GMR] definition, which allows 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 7.16 *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 7.17 (weak zero knowledge [DOY]) *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. (An intermediate notion is that of ε-knowledge [DNS], where the simulator's running time is required to be bounded by a fixed polynomial in $p(n)$ and $t(n)$.)

We remark that in the past, expected polynomial-time simulators and weak simulators have arisen mainly when considering cheating verifiers (*e.g.,* in [GMR, GMW, GK1, DOY, DNS]); that is, *strict* polynomial-time simulators have always seemed to suffice for simulating the *honest* verifier's view. For such cases, an equivalence between zero knowledge with weak simulators (for cheating verifiers) and zero knowledge with strict polynomial-time simulators has already been established by our result that **HVZK** = **ZK** (Theorem 6.1). However, this does leave open the possibility that weak simulation makes a difference for honest-verifier zero knowledge. We rule out this possibility in the following theorem.

Theorem 7.18 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 Condition, and applying Theorem 6.1. (By definition, **weak-HVZK** $\subseteq$ **IP**.) We will do this by extending our proof that every problem in **HVZK** satisfies the SZK/OWF Condition (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 [Gol4, Thm. 2.3.2]).

We begin with an extension of Lemma 3.7.

Lemma 7.19 *If a promise problem* Π *is in* **weak-HVZK***, then* Π *satisfies the following* Weak Conditional Pseudoentropy Condition*: 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).[16] Note, however, that we allow the sizes of the circuits and their input length (*i.e.*, number of coin tosses) can indeed depend on p.

Proof Sketch: Recall that the proof of Lemma 3.7 first constructed distributions X and 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 for any given polynomial p, we take S to be the simulator achieving ε-indistinguishability, where $\varepsilon(|x|) = 1/(\ell(|x|) \cdot p(|x|))$.

As in the proof of Lemma 3.7, 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 $\mathrm{H}(X'|Y') = r/\ell$. On the other hand, when $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq (r-1)/\ell$, exactly as in Lemma 3.7.

The final distributions are taken to be $(X_1,\ldots,X_\ell)$ and $(Y_1,\ldots,Y_\ell)$ 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 $\ell \cdot \varepsilon < 1/p$. 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 depend on the choice of p). □

Given this lemma, we proceed to reduce the Weak Conditional Pseudoentropy Condition to the SZK/OWF Condition, analogously to Lemma 3.10. In the proof, we will need a weak analogue of the notion of a false entropy generator:

[16] Indeed, otherwise every promise problem would trivially satisfy the Weak Conditional Pseudoentropy Condition. Let $p = p(n)$ be an arbitrary polynomial, and let $m = p$. Given an input x of length n, let (X,Y) be the distribution that always outputs $(0^m, 0^m)$, let $r = 1$, and let (X',Y') equal $(0^m,0^m)$ with probability $1-1/p$ and equal $(U_m, 0^m)$ with probability $1/p$. Then $\mathrm{H}(X|Y) = 0 \leq r-1$, (X',Y') is $1/p$-close to (X,Y), and $\mathrm{H}(X'|Y') \geq (1/p) \cdot m = r$.

Definition 7.20 *We say that there is an* auxiliary-input weak false entropy generator on I *if there exists a* fixed *polynomial m such that for* every *polynomial p, we have samplable auxiliary-input probability ensembles $\mathcal{D} = \{D_x\}$ and $\mathcal{F} = \{F_x\}$ such that D_x and F_x take values in $\{0,1\}^{m(|x|)}$ and when $x \in I$, D_x and F_x are $1/p(|x|)$-indistinguishable and satisfy* $\mathrm{H}(F_x) \geq \mathrm{H}(D_x) + 1$.

The following generalization of Lemma 3.12, proven in Appendix B, states such weak false entropy generators also imply one-way functions.

Lemma 7.21 *If there is an auxiliary-input weak false entropy generator on I, then there exists an auxiliary-input one-way function on I.*

We now use this to establish the SZK/OWF Condition.

Lemma 7.22 *If a promise problem satisfies the* Weak Conditional Pseudoentropy Condition*, then it satisfies the* SZK/OWF Condition.

Proof: Let Π be a promise problem satisfying the Weak Conditional Pseudoentropy Condition, with m being the associated fixed polynomial. Then for any given $\varepsilon = \varepsilon(n) = 1/\mathrm{poly}(n)$ and any instance $x \in \{0,1\}^n$, we can efficiently construct two samplable distributions (X, Y) on $\{0,1\}^m \times \{0,1\}^m$ and a parameter r such that if $x \in \Pi_Y$, then $\mathrm{H}(X'|Y') \geq r + 1$ for some (X', Y') that is ε-indistinguishable from (X, Y), and if $x \in \Pi_N$, then $\mathrm{H}(X|Y) \leq r - 1$.

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 to the one given in the proof of Lemma 3.10.

Thus, we focus on constructing one-way functions on I. The first step of the construction (given in the proof of Lemma 3.10) does not change. We set $k = 4n \cdot (m+n)^2$, and consider the samplable distributions

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

As in the proof of Lemma 3.10, $\mathrm{H}(F) \geq \mathrm{H}(D) + 1$. The only change is that instead of arguing that D and F are computationally indistinguishable, we claim that they are ε'-indistinguishable from Z for $\varepsilon' = 2k \cdot \varepsilon$. The deterioration by a factor of k comes from applying the hybrid argument to k samples of (X_i, Y_i); this occurs both when relating D to D^* and when relating F to F^* in the proof of Lemma 3.10, hence the additional factor of 2. 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 (by a suitable choice of ε). Moreover, the output length m' of D and F remain independent of the choice of $\varepsilon' = 1/\mathrm{poly}(n)$. Thus we have a weak auxiliary-input false entropy generator on I. By Lemma 7.21, we have an auxiliary-input one-way function on I, as needed. ■

8 Open Problems

There are some results that are known about **ZK** under the assumption that one-way functions exist, but for which we have not given unconditional proofs:

1. **ZK** is closed under complement. (If one-way functions exist, then **ZK** = **IP** = **PSPACE** = **co-PSPACE**. [GMW, IY, BGG+, Nao, HILL, LFKN, Sha2])

2. 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].)

3. 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 an **NP** witness for membership [GMW]. (In our Theorem 6.1, the prover needs an **NP** *oracle*.)

The only bottleneck for proving the latter two results unconditionally is our instance-dependent commitment scheme for **SZK** (Theorem 4.3), which has polynomially many rounds and a $\mathbf{BPP}^{\mathbf{NP}}$ sender, so any improvement to that commitment scheme in these respects would have an analogous impact on **ZK**. In fact, at the time of this work, the last two items (round complexity and prover efficiency) were open problems for **SZK** as well, and in [MV] instance-dependent commitments were proposed as an approach to the question of prover efficiency for **SZK**. Subsequent to this work, in joint work with Minh Nguyen [NV], we have resolved the prover efficiency question, proving Item 3 unconditionally, as well as its **SZK** analogue. That work does not, however, construct standard instance-dependent commitment schemes with an efficient sender for all of **SZK** and **ZK** (but rather some new variant of such commitment schemes), and this remains an interesting open problem having additional consequences, e.g. for unconditional results on concurrent zero knowledge [MOSV].

Given that we have been able to prove unconditional results about **ZK**, which allows for computational security in the zero-knowledge condition, a natural next project is to try and handle computational security in the *soundness* condition. That is, undertake a similar unconditional study of zero-knowledge *arguments*, as defined in [BCC, Gol4].

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, and Shien Jin Ong for clarifying discussions. Their comments, as well as those of the anonymous FOCS and SICOMP referees, also improved the presentation significantly. Finally, I 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. *Journal of Computer and System Sciences*, 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. *Journal of Computer and System Sciences*, 36:254–276, 1988.

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

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

[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.

[Blu] M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians (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. *Journal of Computer and System Sciences*, 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.

[Dam1] I. B. Damgård. On the Existence of Bit-Commitment Schemes and Zero-Knowledge Proofs. In G. Brassard, editor, *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 17–29. Springer-Verlag, 1990, 20–24 Aug. 1989.

[Dam2] I. B. Damgård. Interactive Hashing Can Simplify Zero-Knowledge Protocol Design without Computational Assumptions (Extended Abstract). In D. R. Stinson, editor, *Advances in Cryptology—CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 100–109. Springer-Verlag, 22–26 Aug. 1993.

[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 (available from authors), 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.

[DNS] C. Dwork, M. Naor, and A. Sahai. Concurrent zero-knowledge. *Journal of the ACM*, 51(6):851–898 (electronic), 2004.

[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. *Information Processing Letters*, 34(6):277–281, 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. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1999.

[Gol4] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[Gol5] 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 Journal on Computing*, 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. *Journal of the Association for Computing Machinery*, 38(3):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 M. Wiener, editor, *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 467–484. Springer-Verlag, 15–19 Aug. 1999.

[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.

[GMR] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 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). In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, 15–17 May 1989.

[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 Computer Science*, 43(2-3):169–188, 1986.

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

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

[MOSV] D. Micciancio, S. J. Ong, A. Sahai, and S. Vadhan. Concurrent Zero Knowledge without Complexity Assumptions. In S. Halevi and T. Rabin, editors, *Proceedings of the Third Theory of Cryptography Conference (TCC '06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 4–7 March 2006.

[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.

[NV] M. Nguyen and S. Vadhan. Zero Knowledge with Efficient Provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, 21–23 May 2006. To appear.

[NT] N. Nisan and A. Ta-Shma. Extracting Randomness: A Survey and New Constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

[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.

[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*, pages 3–17, 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.

[Sha1] R. Shaltiel. Recent Developments in Extractors. In *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, volume I: Algorithms and Complexity. World Scientific, 2004.

[Sha2] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, Oct. 1992.

[Sip] M. Sipser. A Complexity Theoretic Approach to Randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 Apr. 1983.

[Sto] L. Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985.

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

[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.

[Vad3] S. P. Vadhan. An Unconditional Study of Computational Zero Knowledge. *SIAM Journal on Computing*, 2006. Special Issue on Randomness and Complexity. To appear.

[Yao] A. C. Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 Nov. 1982. IEEE.

A Lemmas about Flat Distributions

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 $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$, 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 $\overline{X} = (X_1, \ldots, X_k)$ and $\overline{Y} = (Y_1, \ldots, Y_k)$, we have:

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

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 $t\Phi/k$. By the Hoeffding Inequality, this probability is at most

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

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

Lemma A.2 (Claim 5.9, restated) *Let Z_0 and Z_1 be Φ-flat distributions, for $\Phi \geq 1$. Let $Z = Z_C$, where C is a uniformly chosen random bit. Then Z is 3Φ-flat.*

Proof: We need to show that, for every $t \geq 1$, a random sample $z \leftarrow Z$ is not $t \cdot 3\Phi$-typical for Z with probability at most 2^{-t^2}. For this, it suffices to separately bound the probabilities that z is not $t \cdot 3\Phi$-light and that z is not $t \cdot 3\Phi$-heavy. Note that $t \cdot 3\Phi \geq 2t \cdot \Phi + 1$, so we can bound the probabilities with respect to a lightness/heaviness threshold of $2t \cdot \Phi + 1$ instead.

Bounding the lightness probability is relatively straightforward, because z being light for Z implies that it is light for both Z_0 and Z_1. Specifically, for any z that is $(2t \cdot \Phi + 1)$-light for Z, we have

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

Similarly for Z_1. Therefore any such z is also $2t\Phi$-light for Z_0 and Z_1. Hence, if $z \leftarrow Z$, then z is $(2t \cdot \Phi + 1)$-light for Z with probability at most $2^{-(2t)^2}$.

The heaviness probability is a bit more subtle, because z being heavy for Z only implies that it is heavy for one of Z_0 and Z_1: Specifically, if z is $(2t \cdot \Phi + 1)$-heavy for Z, then

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

Thus any such z is $2t\Phi$-heavy for either Z_0 or Z_1. Wlog say that z is heavy for Z_0. The probability that Z_0 outputs a string that $2t\Phi$-heavy (for Z_0) is at most $2^{-(2t)^2}$, by Φ-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 $2t\Phi$-heavy for Z_0. The total probability mass of H_0 under Z_0 is at least $|H_0| \cdot 2^{-\mathrm{H}(Z_0)+2t\Phi}$ and at most $2^{-(2t)^2}$ by Φ-flatness. Thus, $|H_0| \leq 2^{-(2t)^2} \cdot 2^{\mathrm{H}(Z_0)-2t\Phi}$. Then

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

We can do an identical analysis for the strings H_1 that are $2t\Phi$-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 a random sample of Z is not $(2t\Phi + 1)$-typical for Z is at most $2^{-(2t)^2} + 3 \cdot 2^{-(2t)^2} \leq 2^{-t^2}$, for $t \geq 1$. ■

B Weak False Entropy Generators Imply One-Way Functions

We recall the definition of a weak false entropy generator.

Definition B.1 (Definition 7.20, restated) *We say that there is an* auxiliary-input weak false entropy generator on I *if there exists a* fixed *polynomial m such that for* every *polynomial p, we have samplable auxiliary-input probability ensembles $\mathcal{D} = \{D_x\}$ and $\mathcal{F} = \{F_x\}$ such that D_x and F_x take values in $\{0,1\}^{m(|x|)}$ and when $x \in I$, D_x and F_x are $1/p(|x|)$-indistinguishable and satisfy* $\mathrm{H}(F_x) \geq \mathrm{H}(D_x) + 1$.

The following generalizes Lemma 3.12 (due to [HILL]). Intuitively, the weakness of the false entropy generator translates to constructing only a weak one-way function (where the inversion probability is at most, say, 1/2), which is known to imply standard one-way functions [Yao] (cf. [Gol4]).

Lemma B.2 (Lemma 7.21, restated) *If there is an auxiliary-input weak false entropy generator on I, then there exists an auxiliary-input one-way function on I.*

Proof: Let $x \in I$, $n = |x|$, and let $D = D_x$ and $F = F_x$ be the samplable auxiliary-input probability ensembles on $\{0,1\}^m$ that are ε-indistinguishable. Recall that the definition of auxiliary-input weak false entropy generators gives us a fixed polynomial $m = m(n)$ such that we can take $\varepsilon = 1/p(n)$ for any desired polynomial p (which we will choose later in the proof). To construct an auxiliary-input one-way function, we will essentially follow 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 D, we need to ensure that the level of indistinguishability deteriorates only as a function of the output length m of D and the security parameter (but not the number of random bits used to generate D).

This part of the construction depends on "guess" e for (an approximation to) the entropy of D. (At the end we will enumerate over all choices for e.) Specifically, set $k = 256n \cdot (m+n)^2$, let q be the number of random bits used to generate D, let G be a random universal hash function mapping $\{0,1\}^{kq}$ to $\{0,1\}^{kq-ke-k/8}$, and consider the following samplable distributions:

$$\begin{aligned} W_e &= (D(R_1), \ldots, D(R_k), G, G(R_1, \ldots, R_k)), \text{ and} \\ W'_e &= (F_1, \ldots, F_k, G, U_{kq-ke-k/8}), \end{aligned}$$

where $R_1, \ldots, R_k$ are independent copies of U_q, and $F_1, \ldots, F_k$ are independent copies of F.

Claim B.3 *For* $\mathrm{H}(D) \le e \le \mathrm{H}(D) + 1/2$*, we have:*

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

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 $\delta - (k+2) \cdot 2^{-n}$ (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 $\delta = (k+2) \cdot 2^{-n} + k\varepsilon \le 1/2$, for a sufficiently large choice of the polynomial p (recalling that $\varepsilon = 1/p$), and is thus a weak one-way function. Since we do not know the value of $\mathrm{H}(D)$, we consider the function $f_x(y_1, \ldots, y_{2m}) = (W_{1/2}(y_1), W_1(y_2), \ldots, W_{m-1/2}(y_{2m-1}), W_m(y_{2m}))$, 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 [Yao] (cf., [Gol4]) completes the proof. Thus, all that remains is to establish Claim B.3.

Proof of Claim B.3: It will first be useful to remove low-probability samples from both D and F, analogously to Lemma 2.2. Let

$$L = \{z : \Pr[D = z] \le 2^{-n} \cdot 2^{-m}\}.$$

By a union bound, $\Pr[D \in L] \le 2^{-n}$. Then $\hat{D} = D|_{D \notin L}$ is 2^{-n}-close to D and moreover for every $z \in \mathrm{Supp}(\hat{D})$,

$$\Pr\left[\hat{D} = z\right] \ge \Pr[D = z] \ge 1/2^{m+n}.$$

By Lemma 2.1, we have $|\mathrm{H}(\hat{D}) - \mathrm{H}(D)| \le 2^{-n} \cdot m + \mathrm{H}_2(2^{-n})$, which is negligible. By the Flattening Lemma, $\otimes^k \hat{D}$ is Φ-flat for $\Phi = \sqrt{k} \cdot (m+n)$. Analogously, using F we can define a set L' of light samples, and obtain a $\hat{F}$ satisfying the same conclusions.

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

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

Since $\otimes^k D$ and $\otimes^k \hat{D}$ are $k \cdot 2^{-n}$-close (by Lemma 2.3), the same holds with probability at least $1 - (k+1) \cdot 2^{-n}$ over $\overline{z} \leftarrow \otimes^k D$. For any such $\overline{z}$, we have

$$\begin{aligned}
&\#\{(r_1, \ldots, r_k) : \forall i \ D(r_i) = z_i\} \\
&\quad = \ 2^{kq} \cdot \Pr[\otimes^k D = \overline{z}] \\
&\quad \geq \ 2^{kq} \cdot \Pr[\otimes^k D = \overline{z} | \otimes^k D \in (L^c)^k] \cdot \Pr\left[\otimes^k D \in (L^c)^k\right] \qquad (\text{where } L^c = \{z : z \notin L\}) \\
&\quad \geq \ 2^{kq} \cdot \Pr[\otimes^k \hat{D} = \overline{z}] \cdot (1 - k \cdot 2^{-n}) \\
&\quad \geq \ 2^{kq} \cdot 2^{-\sqrt{n}\cdot\Phi - k\cdot\mathrm{H}(\hat{D})} \cdot (1 - k \cdot 2^{-n}) \\
&\quad \geq \ 2^{kq - ke - k/8 + 2n},
\end{aligned}$$

where L^c denotes the complement of L and in the last inequality we use the fact that $\mathrm{H}(\hat{D}) \leq \mathrm{H}(D) + \mathrm{neg}(n) \leq e + \mathrm{neg}(n)$ and $\sqrt{n} \cdot \Phi = k/16$, $2n + 1 \leq k/16$. for sufficiently large n. This implies that conditioned on $(D(R_1), \cdots, D(R_k)) = \overline{z}$, the min-entropy of $(R_1, \ldots, R_k)$ is at least $kq - ke - k/8 + 2n$. Thus, by the Leftover Hash Lemma (Lemma 2.7), $(G, G(R_1, \ldots, R_k))$ is 2^{-n}-close to $(G, U_{kq-ke-k/8})$. We conclude that W_e is statistically indistinguishable from

$$V = (D_1, \ldots, D_k, G, U_{kq-ke-k/8}),$$

where $D_1, \ldots, D_k$ are independent copies of D. Since D is ε-indistinguishable from F, 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 $kq + g$. Hence $|\mathrm{Supp}(W_e)| \leq 2^{kq+g}$. Next show that W'_e is statistically indistinguishable from a distribution with min-entropy significantly higher than $kq + g$. This amounts to lower-bounding the min-entropy of $(F_1, \ldots, F_k) = \otimes^k F$, since the remaining components of the W'_e are independent and have min-entropy $g + kq - ke - k/8$. As above, instead of F, we consider $\hat{F}$. Recall that $\otimes^k \hat{F}$ is $(k2^{-n})$-close to $\otimes^k F$, and is Φ-flat. By Φ-flatness, $\otimes^k \hat{F}$ is 2^{-n}-close to a distribution with min-entropy $k \cdot \mathrm{H}(\hat{F}) - \sqrt{n}\Phi \geq k \cdot (e + 1/2 - \mathrm{neg}(n)) - k/16 \geq ke + k/4$ for sufficiently large n. Therefore, W'_e is $(k+1) \cdot 2^{-n}$-close to a distribution with min-entropy at least

$$(g + kq - ke - k/8) + (ke + k/4) > kq + g + n$$

for sufficiently large n. A distribution of min-entropy at least $w = kq + g + n$ 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. □