Interactive and Noninteractive Zero Knowledge Coincide in the Help Model[*]

Dragos Florin Ciocan and Salil Vadhan[†]
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
ciocan@post.harvard.edu, salil@eecs.harvard.edu

October 3, 2007

Abstract

We show that a problem in AM has a interactive zero-knowledge proof system *if and only if* it has a noninteractive zero knowledge proof system in the 'help model' of Ben-Or and Gutfreund (*J. Cryptology*, 2003). In this model, the shared reference string is generated by a probabilistic polynomial-time dealer who is given access to the statement to be proven. Our result holds for both computational zero knowledge and statistical zero knowledge, and does not rely on any unproven complexity assumptions.

We also show that help does not add power to interactive computational zero-knowledge proofs, paralleling a result of Ben-Or and Gutfreund for the case of statistical zero knowledge.

Keywords: cryptography, computational complexity, noninteractive zero-knowledge proofs, commitment schemes, Arthur–Merlin games, one-way functions

[*] Some of this work previously appeared in the first author's undergraduate thesis [Cio].

[†] Supported by NSF Grant CNS-0430336 and a Guggenheim Fellowship.

1 Introduction

Zero-knowledge proofs [GMR] are protocols whereby a prover can convince a verifier that some assertion is true with the property that the verifier learns nothing else from the protocol. This remarkable property is easily seen to be impossible for the classical notion of a proof system, where the proof is a single string sent from the prover to the verifier, as the proof itself constitutes 'knowledge' that the verifier could not have feasibly generated on its own (assuming NP $\not\subseteq$ BPP). Thus zero-knowledge proofs require some augmentation to the classical model for proof systems.

The original proposal of Goldwasser, Micali, and Rackoff [GMR] augments the classical model with both randomization and multiple rounds of interaction between the prover and the verifier, leading to what are called *interactive zero-knowledge proofs*, or simply *zero-knowledge proofs*. An alternative model, proposed by Blum, Feldman, and Micali [BFM, BDMP], augments the classical model with a set-up in which a trusted dealer who randomly generates a *reference string* that is shared between the prover and verifier. After this reference string is generated, the proof is consists of just a single message from the prover to verifier. Thus, these are referred to *noninteractive zero-knowledge proofs*. Since their introduction, there have been many constructions of both interactive and noninteractive zero-knowledge proofs, and both models have found wide applicability in the construction of cryptographic protocols.

It is natural to ask what is the relation between these two types of models. That is:

> Can every assertion that can be proven with an interactive zero-knowledge proof also be proven with a noninteractive zero-knowledge proof, and conversely?

Our main result is a positive answer to this question in the 'help model' of Ben-Or and Gutfreund [BG], where the dealer is given access to the statement to be proven when generating the reference string. We hope that this will provide a step towards answering the above question for more standard models of noninteractive zero knowledge, such as the common reference string model and the public parameter model.

1.1 Models of Zero Knowledge

Interactive Zero Knowledge. Recall that an *interactive proof system* [GMR] for a problem Π is an interactive protocol between a computationally unbounded prover P and a probabilistic polynomial-time verifier V that satisfies the following two properties:

- *Completeness:* if x is a YES instance of Π, then the V will accept with high probability after interacting with the P on common input x.
- *Soundness:* if x is a NO instance of Π, then for every (even computationally unbounded) prover strategy P^*, V will reject with high probability after interacting with P^* on common input x.

Here, we consider problems Π that are not only languages, but also ones that are *promise problems*, meaning that some inputs can be neither YES nor NO instances, and we require nothing of the protocol on such instances. (Put differently, we are 'promised' that the input x is either a YES or a NO instance.) We write IP for the class of promise problems possessing interactive proof systems.

As is common in complexity-theoretic studies of interactive proofs and zero knowledge, we allow the honest prover P to be computationally unbounded, and require soundness to hold against computationally unbounded provers. However, cryptographic applications of zero-knowledge proofs typically require an honest prover P that can be implemented in probabilistic polynomial-time given a witness of membership for x, and it often suffices for soundness to hold only for polynomial-time prover strategies P^* (leading to *interactive argument systems* [BCC])). It was recently shown how to extend the complexity-theoretic studies of interactive zero knowledge proofs to both polynomial-time honest provers [NV], and to argument systems [OV2]; we hope that the same will eventually happen for noninteractive zero knowledge.

Intuitively, we say that an interactive proof system is *zero knowledge* if the verifier 'learns nothing' nothing from the interaction other than the fact that the assertion being proven is true, even if the verifier deviates from the specified protocol. Formally, we require that there is an efficient algorithm, called the *simulator*, that can simulate the verifier's view of the interaction given only the (YES) instance x and no access to the prover P. The most general notion, *computational zero knowledge* or just *zero knowledge*, requires this to hold for all polynomial-time cheating verifier strategies (and the simulation should be computationally indistinguishable from the verifier's view). A stronger notion, *statistical zero knowledge*, requires security against even computationally unbounded verifier strategies (and the simulation should be statistically indistinguishable from the verifier's view). We write ZK (resp., SZK) to denote the class of promise problems possessing computational (resp., statistical) zero-knowledge proof systems. [1]

Noninteractive Zero Knowledge. For noninteractive zero knowledge [BFM, BDMP], we introduce a trusted third party, the *dealer*, who randomly generates a *reference string* that is provided to both the prover and verifier. After that, the prover noninteractively sends a single message to the verifier, who decides whether to accept or reject. Completeness and soundness are defined analogously to interactive proofs, except that the probabilities are now also taken over the choice of the reference string. Computational and statistical zero knowledge are also defined analogously to the interactive case, except that now the reference string is also considered part of the verifier's view, and also must be simulated. (In this paper, we consider *single-theorem*, *nonadaptive* zero knowledge, where the zero-knowledge property is guaranteed provided only one statement is proven, and this statement is independent of the shared reference string. Some cryptographic applications require *many-theorem* zero knowledge, where polynomially many statements can be proven in zero knowledge using the same reference string, and/or *adaptive* zero knowledge, where a cheating verifier may choose the statement(s) after seeing the reference string.)

There are a number of variants of the noninteractive model, depending on the form of the trusted set-up performed by the dealer. In the original, *common random string (crs) model* proposed by Blum et al. [BFM, BDMP], the reference string is simply a uniformly random string of polynomial length. This gives rise to the classes $\mathrm{NIZK}^{\mathrm{crs}}$ and $\mathrm{NISZK}^{\mathrm{crs}}$ of problems having noninteractive computational and statistical zero-knowledge proofs in the common random string model. A natural and widely used generalization is the *public parameter model*, where the reference string need not be uniform, but can be generated according to any polynomial-time samplable distribution. That is, we obtain the reference string by running a probabilistic polynomial-time *dealer* algorithm D on input 1^n, where n is the length of statements to be proven (or the security parameter). This model gives rise to the classes $\mathrm{NIZK}^{\mathrm{pub}}$ and $\mathrm{NISZK}^{\mathrm{pub}}$.

[1] In some papers, such as [OV2, OV1], a prefix of C is used to denote *computational* zero knowledge and a suffix of P is used to specify interactive *proof* systems rather than arguments, so ZK and SZK would be CZKP and SZKP, respectively. We opt for more streamlined notation here for readability.

A further generalization is the *help model* introduced by Ben-Or and Gutfreund [BG]. In this model, the distribution of the reference string is allowed to depend on the statement x being proven. That is, the reference string is generated by running a probabilistic polynomial-time dealer algorithm D on input x. We denote the class of problems having computational (resp. statistical) zero-knowledge proofs in this model as $\mathrm{NIZK}^{\mathrm{h}}$ (resp., $\mathrm{NISZK}^{\mathrm{h}}$). This model does not seem to suffice for most cryptographic applications, but its study may serve as a stepping stone to a better understanding of the more standard models of noninteractive zero knowledge mentioned above. Indeed, any characterizations of noninteractive zero knowledge in the help model already serve as *upper bounds* on the power of noninteractive zero knowledge in the common random string and public parameter models.

We remark that one can also consider protocols in which we allow both a trusted dealer and many rounds of interaction. The most general model allows both help and interaction, yielding the classes ZK^{h} and $\mathrm{SZK}^{\mathrm{h}}$, to which we will refer later.

1.2 Previous Work

Recall that we are interested in the relationship between the interactive zero-knowledge classes ZK and SZK and their various noninteractive counterparts, which we will denote by NIZK and NISZK when we do not wish to specify the model. That is, for a given model of noninteractive zero knowledge, we ask: Does ZK = NIZK and does SZK = NISZK?

ZK vs. NIZK. A first obstacle to proving equality of ZK and NIZK is that NIZK is a subset of AM, the class of problems having constant-round interactive proof systems [BM, GS], whereas ZK may contain problems outside of AM. So, instead of asking whether ZK = NIZK, we should instead ask if $\mathrm{ZK} \cap \mathrm{AM} = \mathrm{NIZK}$.

Indeed, this equality is known to hold under complexity assumptions. If one-way permutations exist, then it is known that ZK = IP [GMW, IY, BGG^{+}] and $\mathrm{NIZK}^{\mathrm{crs}} = \mathrm{AM}$ [FLS], and thus $\mathrm{ZK} \cap \mathrm{AM} = \mathrm{NIZK}^{\mathrm{crs}} = \mathrm{NIZK}^{\mathrm{pub}} = \mathrm{NIZK}^{\mathrm{h}}$. (In fact, if we replace $\mathrm{NIZK}^{\mathrm{crs}}$ with $\mathrm{NIZK}^{\mathrm{pub}}$, these results hold assuming the existence of any one-way function [HILL, Nao, GB, Pas].)

Thus, for computational zero knowledge, the interesting question is whether we can prove that $\mathrm{ZK} \cap \mathrm{AM} = \mathrm{NIZK}$ *unconditionally*, without assuming the existence of one-way functions. To our knowledge, there have been no previous results along these lines.

SZK vs. NISZK. For relating SZK and NISZK, the class AM no longer is a barrier, because it is known that $\mathrm{SZK} \subseteq \mathrm{AM}$ [AH].

The relationship between SZK and NISZK was first addressed in the work of Goldreich et al. [GSV2]. There it was shown that SZK and $\mathrm{NISZK}^{\mathrm{crs}}$ have the 'same complexity' in the sense that SZK = BPP iff $\mathrm{NISZK}^{\mathrm{crs}} = \mathrm{BPP}$. Moreover, it was proven that $\mathrm{SZK} = \mathrm{NISZK}^{\mathrm{crs}}$ iff $\mathrm{NISZK}^{\mathrm{crs}}$ is closed under complement.

In addition to introducing the help model, Ben-Or and Gutfreund [BG] studied the relationship between $\mathrm{NISZK}^{\mathrm{h}}$ and SZK. They proved that $\mathrm{NISZK}^{\mathrm{h}} \subseteq \mathrm{SZK}$ (in fact that $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$), and posed as an open question whether $\mathrm{SZK} \subseteq \mathrm{NISZK}^{\mathrm{h}}$.[2]

[2] In fact, their conference paper [GB] claimed to prove that $\mathrm{SZK} = \mathrm{NISZK}^{h}$, but this was retracted in the journal version [BG].

1.3 Our Results.

We show that interactive zero knowledge does in fact collapse to noninteractive zero knowledge in the help model, both for the computational case (restricted to AM) and the statistical case:

Theorem 1.1 $\mathrm{ZK} \cap \mathrm{AM} = \mathrm{NIZK}^{\mathrm{h}}$.

Theorem 1.2 $\mathrm{SZK} = \mathrm{NISZK}^{\mathrm{h}}$.

These results and their proofs yield new characterizations of the classes ZK and SZK. For example, we obtain a new complete problem for SZK, namely the $\mathrm{NISZK}^{\mathrm{h}}$-complete problem given in [BG]. Similarly, we obtain a new characterization of ZK, which amounts to a computational analogue of the $\mathrm{NISZK}^{\mathrm{h}}$-complete problem. As suggested in [BG], these results can also be viewed as first steps towards collapsing interactive zero knowledge to noninteractive zero knowledge in the public parameter or common reference string model. For example, to show $\mathrm{SZK} = \mathrm{NISZK}^{\mathrm{crs}}$ (the question posed in [GSV1]), it now suffices to show that $\mathrm{NISZK}^{\mathrm{h}} = \mathrm{NISZK}^{\mathrm{crs}}$.

As mentioned above, one can consider even more general classes ZK^{h} and $\mathrm{SZK}^{\mathrm{h}}$ that incorporate both help and interaction. Ben-Or and Gutfreund [BG] showed that $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$. We prove an analogous result for computational zero knowledge:

Theorem 1.3 $\mathrm{ZK}^{\mathrm{h}} = \mathrm{ZK}$.

1.4 Techniques

The main tool we use in showing that interactive zero knowledge collapses to noninteractive zero knowledge in the help model, i.e. $\mathrm{ZK} \cap \mathrm{AM} \subseteq \mathrm{NIZK}^{\mathrm{h}}$ and $\mathrm{SZK} \subseteq \mathrm{NISZK}^{\mathrm{h}}$, are certain variants of *commitment schemes.* Recall that a commitment scheme is a two-stage interactive protocol between a *sender* and a *receiver.* In the *commit stage*, the sender 'commits' to a secret message m. In the *reveal stage*, the sender 'reveals' m and tries to convince the verifier that it was the message committed to in the first stage. Commitments should be *hiding*, meaning that an adversarial receiver will learn nothing about m in the commit stage, and *binding*, meaning that after the commit stage, an adversarial sender should not be able to successfully reveal two different messages (except with negligible probability). Each of these security properties can be either *computational,* holding against polynomial-time adversaries, or *statistical,* holding even for computationally unbounded adversaries. Commitments are a basic building block for zero-knowledge protocols, e.g. they are the main cryptographic primitive used in the constructions of zero-knowledge proofs for all of NP [GMW] and IP [IY, BGG$^+$].

A relaxed notion is that of *instance-dependent commitment schemes* [BMO, IOS, MV]. Here the sender and receiver are given an instance x of some problem Π as auxiliary input. We only require the scheme to be hiding if x is a YES instance, and only require it to be binding if x is a NO instance. They are a relaxation of standard commitment schemes because we do not require hiding and binding to hold simultaneously. Still, as observed in [IOS], an instance-dependent commitment scheme for a problem $\Pi \in \mathrm{IP}$ suffices to construct zero-knowledge proofs for Π because the constructions of [GMW, IY, BGG$^+$] only use the hiding property for zero knowledge (which is only required on YES instances), and the binding property for soundness (which is only required on NO instances).

The converse was recently shown by Ong and Vadhan [OV1]. That is, instance-dependent commitments not only suffice for zero knowledge, but are also necessary. More precisely, SZK consists exactly of the problems Π having instance-dependent commitments that are statistically hiding on YES instances and statistically binding on NO instances, and ZK consists exactly of the problems $\Pi \in$ IP having instance-dependent commitments that are computationally hiding on YES instances and statistically binding on NO instances.

Our results are obtained in two steps. First, we observe that the interactive instance-dependent commitments constructed for ZK and SZK in [OV1] can be made noninteractive, *provided* we assume that the sender will not deviate from the protocol in the commit phase. Second, we show that such noninteractive, honest-sender instance-dependent commitments suffice for a problem $\Pi \in$ AM to have a $\mathrm{NIZK}^{\mathrm{h}}$ proof system (or $\mathrm{NISZK}^{\mathrm{h}}$ proof system, if the commitments are statistically hiding). Combining these two results with those of [OV1] mentioned above, we deduce that interactive zero knowledge collapses to noninteractive zero knowledge in the help model.

The reverse inclusion in the case of statistical zero knowledge, namely $\mathrm{NISZK}^{\mathrm{h}} \subseteq \mathrm{SZK}$, follows from Ben-Or and Gutfreund [BG], who actually proved the stronger statement that $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$. The case of computational zero knowledge is given by Theorem 1.3. To prove this, we follow the lines of Ben-Or and Gutfreund's proof for the statistical case. They showed how to reduce every problem in $\mathrm{SZK}^{\mathrm{h}}$ to the intersection of the two known complete problems for SZK [SV, GV], and then used the fact that SZK is closed under intersection. ZK is not known to have natural complete problems, but we are able to instead use recent characterizations of ZK that are computational analogues of the SZK-complete problems [Vad].

2 Definitions and Preliminaries

2.1 Notation

We will first introduce some of the basic notation that we will use.

We use capital letters to denote random variables. The notation $x \leftarrow X$ means that x is drawn from the distribution X. We define the *support* of a random variable X as $\mathrm{Supp}(X) = \{x : \Pr[X = x] > 0\}$. 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 $\{0,1\}^m$. If a distribution X can be represented by a circuit which can be described and evaluated in polynomial time, we say X is an *efficiently samplable distribution.*

We use the shorthand PPT for probabilistic polynomial time algorithms. For a PPT A, we write $A(x;r)$ to denote the output of A on input x with randomness r. A *nonuniform* PPT algorithm is a pair $(A, \overline{z})$, where $\overline{z}$ is an infinite series of inputs $z_1, \ldots, z_n, \ldots$ such that $|z_n| = \mathrm{poly}(n)$, and A is a PPT which receives inputs $(x, z_{|x|})$.

A function $\varepsilon : N \to [0,1]$ is called *negligible* if $\varepsilon(n) = n^{-\omega(1)}$. We use $\mathrm{neg}(n)$ to denote an arbitrary negligible function, and $\mathrm{poly}(n)$ to denote an arbitrary polynomial function.

2.2 Promise Problems

Promise problems are a more general variant of decision problems than languages. A promise problem Π is a pair of disjoint sets of strings (Π_Y, Π_N), where Π_Y is the set of YES instances and Π_N is the set of NO instances. The computational problem associated with any promise problem Π is: given a string that is "promised" to lie in $\Pi_Y \cup \Pi_N$, decide whether it is in Π_Y or

Π_N. Reductions from one promise problem to another are natural extensions of reductions between languages. Namely, we say Π reduces to Γ if there exists a polynomial time computable function f such that $x \in \Pi_Y \Rightarrow f(x) \in \Gamma_Y$ and $x \in \Pi_N \Rightarrow f(x) \in \Gamma_N$. We can also naturally extend the definitions of complexity classes by letting the properties of the strings in the languages be conditions on the YES instances, and properties of strings outside of the language be conditions on NO instances.

2.3 Instance-Dependent Cryptographic Primitives

Many of the objects that we will be constructing for use in our zero knowledge constructions will be instance dependent. Hence, we will modify common cryptographic primitives such as one-way functions by allowing them to be parametrized by some string x, such that the cryptographic properties will only be guaranteed to hold if x is in some set I.

Definition 2.1 *An* instance-dependent 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* $p(\cdot)$ *and* $q(\cdot)$ *are polynomials.* $\mathcal{F}$ *is* polynomial-time computable *if there exists a polynomial-time algorithm* F *such that for all* $x \in \{0,1\}^*$ *and* $y \in \{0,1\}^{p(|x|)}$*,* $F(x,y) = f_x(y)$*.*

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

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

Definition 2.3 *An* instance-dependent probability ensemble *on* I *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 *is there exists a probabilistic polynomial-time algorithm* M *such that for every input* x*,* $M(x)$ *is distributed according to* X_x*.*

Definition 2.4 *Two instance-dependent probabilistic ensembles* $\{X_x\}$ *and* $\{Y_x\}$ *are* computationally indistinguishable *on* $I \subset \{0,1\}^*$ *if for every nonuniform PPT* D*, there exists a negligible* $\varepsilon(\cdot)$ *such that for all* $x \in I$*,*

$$\Pr\left[D(x, X_x) = 1\right] - \Pr\left[D(x, Y_x) = 1\right] | \leq \varepsilon(|x|)$$

Similarly, we say $\{X_x\}$ *and* $\{Y_x\}$ *are* statistically indistinguishable *on* $I \subset \{0,1\}^*$ *if the above is required for all functions* D*. If* X_x *and* Y_x *are identically distributed for all* $x \in I$*, we say they are* perfectly indistinguishable *.*

We will sometimes use the informal notation $X \overset{\mathrm{c}}{\equiv} Y$ to denote that ensembles X and Y are computationally indistinguishable.

Definition 2.5 *An* instance-dependent pseudorandom generator *on* I *is a polynomial-time instance-dependent 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*.*

2.4 Probability distributions

In this section, we will define several tools that are useful for analysing properties of probability distributions.

Definition 2.6 *The* statistical difference *between two random variables X and Y taking values in some domain U is defined as:*

$$\Delta(X,Y) \quad = \quad \max_{S \subset U} |\Pr[X \in S] - \Pr[Y \in S]| = \frac{1}{2}\sum_{x \in U} |\Pr[X = x] - \Pr[Y = x]|$$

Lemma 2.7 *For random variables X and Y, if $\delta = \Delta(X,Y)$, then for every $k \in \boldsymbol{N}$, we have:*

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

where $\otimes$ denotes the direct product, i.e., when k independent copies of a random variable are taken.

Definition 2.8 *An ordered pair of distributions (X,Y) is called* α-disjoint *if* $\Pr_{x \leftarrow X}[x \in \mathrm{Supp}(Y)] \leq 1 - \alpha$ *. We call (X,Y)* mutually α-disjoint *if both (X,Y) and (Y,X) are α-disjoint.*

Note that disjointness is a more stringent measure of the disparity between two distributions than statistical difference. If two distributions are α-disjoint, then their statistical difference is at least α. The converse, however, does not hold, since the two distributions could have statistical difference that is negligibly close to 1, yet have identical supports and thus be mutually 0-disjoint.

Definition 2.9 *The* entropy *of a random variable X is* $\mathrm{H}(X) = \mathrm{E}_{x \leftarrow X}\left[\log \frac{1}{\Pr[X=x]}\right]$. *The* conditional entropy of X given Y is

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

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

Definition 2.10 *The relative entropy (Kullback-Liebler distance) between two distributions X, Y is*

$$\mathrm{KL}(X|Y) = \mathop{\mathrm{E}}_{x \leftarrow X}\left[\log \frac{\Pr[X = x]}{\Pr[Y = x]}\right]$$

We denote by $\mathrm{H}_2(p)$ the binary entropy function, which is the entropy of a $\{0,1\}$-valued random variable with expectation p. $\mathrm{KL}_2(p,q)$ denotes the relative entropy between two $\{0,1\}$-value random variables with expectations p and q.

3 Interactive Zero Knowledge

We consider a generalized version of interactive zero knowledge, introduced by Ben-Or and Gutfreund [BG], in which the prover and the verifier have access to a help string output by a dealer algorithm that has access to the statement being proven. We will call this model of interactive zero knowledge the *help model.* Interactive zero-knowledge proofs are a special case of interactive zero-knowledge proofs in the help model.

We denote the three algorithms that make up an interactive zero-knowledge proof in the help model as D, P and V. All three receive as input x, the statement being proven. The dealer selects the help string $\sigma \leftarrow D(x)$ and sends it to P and V. P and V carry out an interactive protocol and, at the end of their interaction, they either output ACCEPT or REJECT. We call the *transcript* the sequence of messages which the triple (D, P, V) computes. $(D, P, V)(x)$ denotes the random variable of the possible outcomes of the protocol, while $\langle D, P, V\rangle(x)$ denotes the verifier's view of the transcripts (where the probability space is over the random coins of D, P and V).

Definition 3.1 ***[BG]*** ($\mathrm{ZK^h}$, $\mathrm{SZK^h}$) *A* zero-knowledge proof system in the help model *for a promise problem* Π *is a triple of probabilistic algorithms* (D, P, V) *(where* D *and* V *are polynomial time bounded), satisfying the following conditions:*

1. *Completeness. For all* $x \in \Pi_Y$, $\Pr\left[(D, P, V)(x) = 1\right] \geq \frac{2}{3}$*, where the probability is taken over the coin tosses of* D, P *and* V*.*

2. *Soundness. For all* $x \in \Pi_N$ *and every prover strategy* P^*, $\Pr\left[(D, P^*, V) = 1\right] \leq \frac{1}{3}$*, where the probability is taken over the coin tosses of* D, P^*, V*.*

3. *Zero Knowledge. There exists a PPT* S *such that the ensembles* $\{\langle D, P, V\rangle)(x)\}_x$ *and* $\{S(x)\}_x$ *are computationally indistinguishable on* Π_Y*.*

If the ensembles are statistically indistinguishable, we call it a statistical zero knowledge proof system in the help model.$\mathrm{ZK^h}$ *(resp.,* $\mathrm{SZK^h}$*) is the class of promise problems possessing zero-knowledge (resp., statistical zero-knowledge) proof systems in the help model.*

If the help string σ *is generated according to* $D(1^{|x|})$*, we call the proof system an* interactive zero-knowledge proof system in the public parameter model*. The corresponding complexity class is* $\mathrm{ZK^{pub}}$ *(resp.,* $\mathrm{SZK^{pub}}$*). If the help string* σ *is generated from the uniform distribution on* $\{0,1\}^{|x|}$*, we call the proof system an* interactive zero-knowledge proof system in the common random string model*. The corresponding complexity class is* $\mathrm{ZK^{crs}}$ *(resp.,* $\mathrm{SZK^{crs}}$*).*

If we remove the dealer's help, the resulting proof system is said to be an interactive zero-knowledge proof system*. The corresponding complexity class is* ZK *(resp.,* SZK*).*

It is simple to show that ZK will be contained in IP, the class of promise problems with interactive proofs:

Lemma 3.2 $\mathrm{ZK^h} \subseteq \mathrm{IP}$*, where* IP *is the class of problems having interactive proofs.*

Proof: We can transform a $\mathrm{ZK^h}$ proof by just having the verifier simulate the dealer's help. This will not preserve zero knowledge in general, since even the honest verifier will learn the dealer's secret coin tosses, but it will preserve completeness and soundness. ■

3.1 Statistical Zero Knowledge

In this section, we state a few characterizations of statistical zero knowledge which will be related to the ones we will later obtain for the computational case. We begin by noting that, in the statistical case, zero knowledge in the help model is equivalent to zero knowledge ([BG]):

Theorem 3.3 ***[BG]*** $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$.

The theorem above implies that all the characterizations of SZK will also hold for $\mathrm{SZK}^{\mathrm{h}}$. In particular, $\mathrm{SZK}^{\mathrm{h}}$ will share the two complete problems for SZK that are due to [GV, SV, Vad]:

Theorem 3.4 ***[SV]*** *The promise problem* Statistical Difference*, defined as:*

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

is complete for SZK*, where X and Y are samplable distributions specified by circuits that sample from them.*

Theorem 3.5 ***[GV, Vad]*** *The promise problem* Conditional Entropy Approximation*, defined as:*

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

is complete for SZK*, where (X,Y) is a joint samplable distribution specified by circuits that use the same coin tosses.*

3.2 Computational Zero Knowledge

In the case of ZK, no natural complete problems are known (unless we assume that one-way functions exist, in which case ZK = IP = PSPACE [GMR, IY, BGG^{+}, Sha, LFKN, HILL, Nao]). However, characterizations that are analogous to the complete problems for SZK do exist in the form of the Indistinguishability Condition and the Conditional Pseudoentropy Condition below. These conditions give 'if and only if' characterizations of ZK that provide essentially the same functionality as complete problems.

The first characterization will be a natural extension of SD to ZK:

Definition 3.6 *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.*
- *If $x \in \Pi_N$, then $\Delta(X,Y) \geq 2/3$.*

Theorem 3.7 ***[Vad]*** $\Pi \in \mathrm{ZK}$ *if and only if* $\Pi \in \mathrm{IP}$ *and* Π *satisfies the* Indistinguishability Condition.

The second characterization is based on the SZK-complete problem CEA:

Definition 3.8 *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) *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* $H(X'|Y') \geq r$*.*
- *If* $x \in \Pi_N$*, then* $H(X|Y) \leq r - 1$*.*

Theorem 3.9 *[**Vad**]* $\Pi \in \mathrm{ZK}$ *if and only if* $\Pi \in \mathrm{IP}$ *and* Π *satisfies the* Conditional Pseudoentropy Condition*.*

4 Noninteractive Zero Knowledge

4.1 The Help Model

In this section, we define a special version of zero-knowledge proofs in the help model, namely noninteractive, zero-knowledge proofs.

Definition 4.1 *[**BG**]* ($\mathrm{NIZK^h}$, $\mathrm{NISZK^h}$) *A* noninteractive zero-knowledge proof system in the help model *for a promise problem* Π *is an interactive zero-knowledge proof in which there is only one message* $\pi = P(x, \sigma)$ *from prover to verifier.*

If the real transcripts are statistically indistinguishable from simulated ones, we call it a noninteractive statistical zero knowledge proof system*.* $\mathrm{NIZK^h}$ *(resp.,* $\mathrm{NISZK^h}$*) is the class of promise problems possessing noninteractive zero-knowledge (resp., noninteractive statistical zero-knowledge) proof systems.*

If the help string σ *is generated according to* $D(1^{|x|})$*, we call the proof system a* noninteractive zero-knowledge proof system in the public parameter model*. The corresponding complexity class is* $\mathrm{NIZK^{pub}}$ *(resp.,* $\mathrm{NISZK^{pub}}$*). If the help string* σ *is generated from the uniform distribution on* $\{0, 1\}^{|x|}$*, we call the proof system an* noninteractive zero-knowledge proof system in the common random string model*. The corresponding complexity class is* $\mathrm{NIZK^{crs}}$ *(resp.,* $\mathrm{NISZK^{crs}}$*).*

Note that the class AM (Definition 4.2) proves to be a natural upper bound for $\mathrm{NIZK^h}$, since we can just have the verifier replace the dealer in creating the reference string. Also, another (lower) bound for $\mathrm{NIZK^h}$ is $\mathrm{NIZK^{crs}}$, which is definitionally a more restricted version of the help model.

The main benefit of the public parameter model and the help model over the simpler CRS model is that they make it easier to construct NIZK proofs from simpler cryptographic primitives such as one-way functions ([BG, Pas]), or, as we will show in this paper, from a certain kind of commitment schemes.

4.2 The Hidden Bits Model

In this section, we define the hidden bits model, which is the basis for our commitment-based construction of $\mathrm{NISZK^h}$ (resp., $\mathrm{NIZK^h}$), and has also proved very useful as a building block for efficient prover constructions of NIZK, such as NIZK proof systems in the CRS model ([FLS]).

We begin by defining a natural bound for noninteractive zero-knowledge in the hidden bits model:

Definition 4.2 (AM) *An* AM proof system *is a pair of probabilistic algorithms (P, V) where the prover P(Merlin) is unbounded, whereas the verifier V (Arthur) is PPT. V sends a random string $r \overset{R}{\leftarrow} \{0,1\}^{\mathrm{poly}(|x|)}$, to which P sends a single response m. V decides then accepts or rejects with no more randomness (i.e. V is a deterministic function of x, r and m). Equivalently, a promise problem $\Pi \in$ AM if $\exists$ a polynomial-time algorithm V, and polynomials $p(|x|), q(|x|)$ such that:*

1. *Completeness.* $x \in \Pi_Y \Rightarrow \Pr_{r \in \{0,1\}^{p(|x|)}}[\exists m \in \{0,1\}^{q(|x|)} s.t.\ V(x, r, m) = 1] \geq 2/3.$
2. *Soundness.* $x \in \Pi_N \Rightarrow \Pr_{r \in \{0,1\}^{p(|x|)}}[\exists m \in \{0,1\}^{q(|x|)} s.t.\ V(x, r, m) = 1] \leq 1/3.$

In other words, the class AM captures all promise problems that have 2-round public coin interactive proof systems. Although AM seems to be a very restrictive class of interactive protocols, it is in fact known that it contains all problems having interactive proof systems with a constant number of rounds, not just 2-round public coin proof systems [BM, GS].

The Hidden Bits Model. The hidden bits model is a fictituous model due to Feige, Lapidot and Shamir [FLS], that allows for an unconditional construction of NIZK. It assumes that both the prover P and the verifier V share a common reference string σ, which we will call the hidden random string (HRS). However, only the prover can see the HRS. We can imagine that the individual bits of σ are locked in boxes, and only the prover has the keys to unlock them. The prover can selectively unlock boxes and reveal bits of the hidden random string. However, without the prover's help, the verifier has no information about any of the bits in the HRS.

Definition 4.3 ***[FLS]*** (***NIZK in the Hidden Bits Model***) *A* noninteractive zero knowledge proof system in the hidden-bits model *for a promise problem* Π *is a pair of probabilistic algorithms (P, V) (where P and V polynomial-time bounded) and a polynomial $l(|x|) = |\sigma|$, satisfying the following conditions:*

1. *Completeness. For all $x \in \Pi_Y$,* $\Pr[\exists (I, \pi) s.t.\ V(x, \sigma_I, I, \pi) = 1] \geq \frac{2}{3}$*, where $(I, \pi) = P(x, \sigma)$, I is a set of indices in $\{0, \ldots, l(k)\}$, and σ_I is the sequence of opened bits of σ, $(\sigma_i : i \in I)$, and where the probability is taken over $\sigma \overset{R}{\leftarrow} \{0,1\}^{l(|x|)}$ and the coin tosses of P and V.*
2. *Soundness. For all $x \in \Pi_N$ and all P^*,* $\Pr[\exists (I, \pi) s.t.\ V(x, \sigma_I, I, \pi) = 1] \leq \frac{1}{3}$*, where $(I, \pi) = P^*(x, \sigma)$, where the probability is taken over $\sigma \overset{R}{\leftarrow} \{0,1\}^{l(|x|)}$ and the coin tosses of P^* and V.*
3. *Zero Knowledge. There exists a PPT S such that the ensembles of transcripts $\{(x, \sigma, P(x, \sigma))\}_x$ and $\{S(x)\}_x$ are statistically indistinguishable on Π_Y, where $\sigma \overset{R}{\leftarrow} \{0,1\}^{l(|x|)}$.*

Note that we have defined the zero-knowledge condition in this model to be statistical rather than computational. Indeed, the known construction of hidden bits NIZK proof systems is unconditional and yields statistically indistinguishable proof systems.

Theorem 4.4 ***[FLS]*** *Every promise problem* $\Pi \in$ NP *has a hidden bits zero knowledge proof system (P, V).*

As has been observed before (e.g. [Pas]), this construction for NP automatically implies one for all of AM.

Corollary 4.5 *Every promise problem* $\Pi \in$ AM *has a hidden bits zero knowledge proof system* (P, V).

Proof: We will show this by transforming an AM proof into a statement that there exists some message from the prover that the verifier accepts. Since this statement is an NP statement, it can be proven in the hidden bits NIZK model.

Consider Π with an AM proof system (P', V'). We can assume that (P', V') have negligible completeness and soundness errors (this can be achieved by a polynomial number of parallel repetitions.) Let $p(|x|)$ be the length of the random challenge that V' sends to P', $q(|x|)$ be the length of V''s message. Consider the following promise problem Γ, which captures the completeness and soundness properties of (P', V'):

$$\begin{aligned}\Gamma_Y &= \{(x,r) : x \in \Gamma, r \in \{0,1\}^{p(|x|)}, \exists \text{ message } m \text{ such that } V'(x,r,m) = 1\}\\ \Gamma_N &= \{(x,r) : x \in \Gamma, r \in \{0,1\}^{p(|x|)}, \nexists \text{ message } m \text{ such that } V'(x,r,m) = 1\}\end{aligned}$$

It is clear that Γ is in NP, so there exists a hidden bits zero knowledge proof system $(P^{''}, V^{''})$ for it. Suppose the length of the hidden string is $l(|x|)$. Because of the vanishing completeness and soundness errors of (P', V'), we know that for a random choice of (x, r), with $x \in \Pi_Y$, the probability $(x, r) \in \Gamma_Y$ is exponentially close to 1. Similarly, if $x \in \Pi_N$ the probability $(x, r) \in \Gamma_N$ is exponentially close to 1.

We can build a hidden bits zero knowledge proof system (P, V) for Π in the following way. We let P and V share a hidden string σ of length $p(|x|) + l(|x|)$. P sets r to the first $p(|x|)$ bits of σ, and reveals them to V. Then, P uses the $l(|x|)$ remaining unrevealed hidden bits of σ to simulate $P^{''}$'s hidden bits proof that $(x, r) \in \Gamma_Y$, and sends this simulated proof to V. V then simulates $V^{''}$ and accepts if and only if $V^{''}$ accepts.

Completeness and soundness follow from the completeness and soundness of (P', V') (as captured by Γ) and of (P'', V''). Finally, the zero knowledge of (P, V) is given by the zero knowledge of $(P^{''}, V^{''})$, and the fact that, for $x \in \Pi_Y$, $(x, r) \in \Gamma_Y$ with high probability ((P', V') has negligible completeness error). In particular, one can construct a simulator S for the proof system (P, V) by randomly selecting an r, and then using the simulator for (P'', V'') to produce proofs that $(x, r) \in \Gamma_Y$. ■

Hence, there exists an unconditional construction of NIZK for all problems in AM. However, this construction holds only in the impractical hidden bits model. In proving our results, we will show how to implement this construction in the help model by exploiting a novel connection to commitment schemes.

5 From ZK to NIZK$^{\mathrm{h}}$

The main result of this section will be that instance-dependent commitment schemes suffice to construct NISZK$^{\mathrm{h}}$/NIZK$^{\mathrm{h}}$ proof systems. First, we will show how to transform interactive commitment schemes into noninteractive ones that can use in a NISZK$^{\mathrm{h}}$/NIZK$^{\mathrm{h}}$ compiler. Then, we will exhibit a construction of such a commitment-based compiler for noninteractive zero knowledge in the help model.

5.1 From ZK to Instance-Dependent Commitments

In this section, we exploit the new results of Ong and Vadhan [OV1], who characterize interactive zero knowledge in terms of a certain kind of commitment scheme. Namely, we will prove that the instance-dependent commitments which characterize SZK/ZK imply the noninteractive, honest-sender, instance-dependent commitment schemes we can use to build $\mathrm{NISZK}^{\mathrm{h}}/\mathrm{NIZK}^{\mathrm{h}}$ proofs.

We begin by defining the commitment schemes that we will be using:

Definition 5.1 *An* instance-dependent commitment scheme *is a family* $\{\mathrm{Com}_x\}_{x\in\{0,1\}^*}$ *with the following properties:*

1. *The scheme* Com_x *proceeds in the stages: the* commit stage *and the* reveal stage. *In both stages, both the* sender *and the* receiver *share as common input the instance* x. *Hence we denote the sender and receiver as* S_x *and, respectively,* R_x, *and we write* $\mathrm{Com}_x = (S_x, R_x)$. *Each party may maintain private state between the two phases.*

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

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

4. *The sender* S_x *and receiver* R_x *algorithms are computable in time* $\mathrm{poly}(|x|)$, *given the instance* x.

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

   An instance-dependent commitment scheme Com_x *is* public coin *if for every* $x \in \{0,1\}^*$, *all of the messages sent by* R_x *in the commit stage are independent random coins, and the receiver maintains no state after the commit stage other than the commitment* c.

We note that, in the definition above, the commit stage may be interactive, requiring the sender and the receiver to exchange messages. The reveal stage, however, is noninteractive, as the receiver only needs the sender's decommitment message to perform its verification.

Security Properties. We now define the security properties of instance-dependent commitment schemes. These properties will be natural extensions of the hiding and binding requirements of standard commitments:

Definition 5.2 *An instance-dependent commitment scheme* $\mathrm{Com}_x = (S_x, R_x)$ *is* statistically (resp., computationally) hiding on $I \subseteq \{0,1\}^*$ *if for every (resp., nonuniform PPT)* R^*, *the ensembles* $\{\mathrm{view}_{R^*}(S_x(0), R^*)\}_{x\in I}$ *and* $\{\mathrm{view}_{R^*}(S_x(1), R^*)\}_{x\in I}$ *are statistically (resp., computationally) indistinguishable, where the random variable* $\{\mathrm{view}_{R^*}(S_x(b), R^*)\}_{x\in I}$ *denotes the view of* R^* *in the commit stage interacting with* $S_x(b)$.

For a promise problem $\Pi = (\Pi_Y, \Pi_N)$, *an instance-dependent commitment scheme* Com_x *is* statistically (resp., computationally) hiding on the YES instances *if* Com_x *is* statistically (resp., computationally) hiding on Π_Y.

Definition 5.3 *An instance-dependent commitment scheme* $\mathrm{Com}_x = (S_x, R_x)$ *is* statistically (resp., computationally) binding on $I \subseteq \{0,1\}^*$ *if for every computationally unbounded (resp., nonuniform PPT)* S^**, there exists a negligible function* ε *such that for all* $x \in I$*, the malicious sender* S^* *succeeds in the following game with probability at most* $\varepsilon(|x|)$*:*

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

For a promise problem $\Pi = (\Pi_Y, \Pi_N)$*, an instance-dependent commitment scheme* Com_x *is* statistically (resp., computationally) binding on the NO instances *if* Com_x *is* statistically (resp., computationally) binding on Π_Y*.*

Ong and Vadhan [OV1] show that the instance-dependent commitment schemes presented above are equivalent to zero knowledge proofs. For the statistical case, [OV1] show that SZK is equivalent to instance-dependent commitments that are statistically hiding on YES instances and binding on NO instances. As expected, the equivalence for ZK is obtained by relaxing the hiding property on YES instances to be computational rather than statistical.

Theorem 5.4 *[OV1] For every promise problem* Π*,* $\Pi \in \mathrm{SZK}$ *if and only if* Π *has an instance-dependent commitment scheme that is statistically hiding on the* YES *instances and statistically binding on the* NO *instances. Moreover, every* $\Pi \in \mathrm{SZK}$ *has an instance-dependent commitment scheme that is public coin and is constant round.*

Theorem 5.5 *[OV1] For every promise problem* Π*,* $\Pi \in \mathrm{ZK}$ *if and only if* Π *has an instance-dependent commitment scheme that is computationally hiding on the* YES *instances and statistically binding on the* NO *instances. Moreover, every* $\Pi \in \mathrm{ZK}$ *has an instance-dependent commitment scheme that is public coin and is constant round.*

5.2 Removing Interaction from Instance-Dependent Commitments

Naturally, we cannot directly use the interactive commitments above in the construction of non-interactive zero-knowledge proof systems. However, in this section we observe that, it is easy to eliminate the interaction between the sender and receiver if we assume that the sender will be honest during the commit stage.

Definition 5.6 *An* noninteractive instance-dependent commitment scheme *is an instance-dependent commitment scheme in which the commit stage consists of a single message* $c = S(x, b)$ *from the sender to the receiver.*

We also redefine the binding property in the context of honest senders:

Definition 5.7 *An instance-dependent commitment scheme* $\mathrm{Com}_x = (S_x, R_x)$ *is* statistically (resp., computationally) binding for honest senders on $I \subseteq \{0,1\}^*$ *if there exists a negligible function* ε *such that for all* $x \in I$*, a computationally unbounded (resp., nonuniform PPT) algorithm* S^* *succeeds in the following game with probability at most* $\varepsilon(|x|)$*:*

> *S interacts with R_x in the commit stage obtaining a commitment c. Then, given the coin tosses of S and the transcript of the commit phase, S^* outputs pairs $(0, d_0)$ and $(1, d_1)$ and* succeeds *if in the reveal stage, $R_x(0, d_0, c) = R_x(1, d_1, c) =$* ACCEPT.

For a promise problem $\Pi = (\Pi_Y, \Pi_N)$, an instance-dependent commitment scheme Com_x *is* statistically (resp., computationally) binding for honest sender of the YES instance *if* Com_x *is* statistically (resp., computationally) binding on Π_Y.

Having defined our noninteractive commitment schemes, we present a transformation from interactive, instance-dependent commitments to noninteractive, instance-dependent commitments for honest senders.

Lemma 5.8 *If a promise problem Π has a public-coin, instance-dependent commitment scheme that is statistically (resp., computationally) hiding on* YES *instances and statistically (resp., computationally) binding for honest senders on* NO *instances, then Π also has a* noninteractive *instance-dependent commitment scheme that is statistically (resp., computationally) hiding on* YES *instances and statistically (resp., computationally) binding for honest senders on* NO *instances.*

Proof: We can obtain a noninteractive, honest-sender, instance-dependent commitment scheme by observing that, if we assume the sender is honest, we can eliminate the interaction between sender and receiver during the commit stage.

If a promise problem Π has an instance-dependent commitment scheme $\mathrm{Com}_x = (S_x, R_x)$, we can modify Com_x to yield a noninteractive, honest-sender, instance-dependent commitment scheme $\mathrm{Com}'_x = (S'_x, R'_x)$ by having the sender carry out the commit phase on its own. Namely, in the commit stage, $S'_x(b)$ will run $(S_x(b), R_x)$ to obtain a commitment c. S'_x will send the commitment c to the receiver. In the reveal stage, S_x simply sends the decommitment d produced by S_x, and R'_x runs R_x (using its commit stage coins r) to verify the decommitment.

The view of R'_x when receiving a commitment c from $S'_x(b)$ is identical to the view of R_x when interacting with $S_x(b)$, so the hiding property of Com_x is maintained. Additionally, since the commitment c output by S'_x is produced by running Com_x, which is statistically (resp., computationally) binding for honest senders on NO instances, and S'_x simply runs (S_x, R_x), it follows that any S^* that succeeds in breaking the binding property of Definition 5.7 for Com'_x will also succeed for Com_x with the same probability. Therefore, Com'_x is also statistically (resp., computationally) binding for honest senders on NO instances. ■

5.3 From Instance-Dependent Commitments to $\mathrm{NIZK}^{\mathrm{h}}$

Theorem 5.9 *If* $\Pi \in \mathrm{AM}$ *and* Π *has a noninteractive, honest-sender, instance-dependent commitment scheme that is statistically (resp., computationally) hiding on* YES *instances and statistically binding for honest senders on* NO *instances, then* $\Pi \in \mathrm{NISZK}^{\mathrm{h}}$ *(resp.,* $\Pi \in \mathrm{NIZK}^{\mathrm{h}}$*).*

Proof: Throughout the proof, we will assume that we have a computationally hiding commitment scheme, which we will use to build a $\mathrm{NIZK}^{\mathrm{h}}$ proof system. The compiler used to build a $NISZKh$ proof system from statistically hiding commitments is identical. We show that we can use a noninteractive, honest-sender, instance-dependent commitment scheme to build a $\mathrm{NIZK}^{\mathrm{h}}$ proof system which implements the hidden bits construction of [FLS] (Definition 4.3 and Theorem 4.5). Our general strategy will be to exploit the correspondence between the algorithms in our definition of an instance-dependent commitment scheme, and the three algorithms in a NIZK$^{\mathrm{h}}$ proof system. More specifically, we will have the dealer D use the sender algorithm to commit to a hidden bits string (this is why we can afford to assume the sender is honest). Since the prover P is allowed to be unbounded, we will use it to exhaustively search for openings to D's commitments. Finally, the verifier V will use the receiver algorithm to check P's openings.

Let $(P^{\mathrm{HB}}, V^{\mathrm{HB}})$ be a hidden bits proof system for Π and let $(\mathrm{Sen}, \mathrm{Rec})$ be the noninteractive, honest-sender bit commitment scheme for Π. We claim that the following proof system (D, P, V) is NIZK$^{\mathrm{h}}$.

1. $D(x, 1^k)$: Select $\sigma^D \xleftarrow{R} \{0,1\}^m$, and run $\mathrm{Sen}(x, \sigma_i^D)$ to generate a commitment c_i, for all i. Output $c = (c_1, \ldots, c_m)$ as the public help parameter.

2. $P(x, c)$: Exhaustively find a random opening o_i^P for each c_i (and, implicitly, each σ_i^D). If one commitment c_i can be opened as both 0 or 1, P outputs o_i^P according to the distribution $O|_{C=c_i}$, where (O, C) is the output of S on a random bit b. Let σ^P be the secret string obtained by P opening D's help string. P runs $P^{\mathrm{HB}}(x, \sigma^P)$ to obtain (I, π). Send $(I, \sigma_I^P, o_I^P, \pi)$ to V.

3. $V(x, I, o_I^P, \pi)$: Compute $\sigma_j^P, \forall j \in I$. Use Rec to check that the commitments are consistent. Run $V^{\mathrm{HB}}(x, I, \sigma_I^P, \pi)$ and accept if and only if V^{HB} accepts.

The reason our protocol refers to 2 secret strings (σ^D and σ^P) is that our commitments are not necessarily binding on YES instances. Consequently, P might not be able to uniquely recover the same secret string σ^D based on D's help string consisting of the commitments to σ^D. That is why we have P recreate another secret string σ^P by drawing from the distribution of bits conditioned on the help string. We note that:

- This only happens for YES instances. For NO instances, P has a negligible chance of being able to open a σ^P different from σ^D. This guarantees that the potential ambiguity of the help string cannot affect soundness.

- The distributions (σ^D, c) and (σ^P, c) are identically distributed (the only difference is the order in which σ and c are drawn).

We now show the protocol described above satisfies the conditions necessary for it to be a NIZK$^{\mathrm{h}}$ proof system:

1. *Completeness.* This follows from the completeness of the hidden bits system $(P^{\mathrm{HB}}, V^{\mathrm{HB}})$.

2. *Soundness.* We show that a potentially malicious prover P^* can open σ^D in only one way with overwhelming probability. Since the commitment scheme is statistically binding on NO instances, the probability that a commitment c_i can be opened as both 0 and 1 will be some negligible function $\varepsilon(n)$, where $n = |x|$. Hence, the probability that any commitment c_i can be opened in two ways is at most $m \cdot \varepsilon(n)$. Assuming that there existed a cheating P^* that could convince V to accept with probability p, then we can obtain a cheating $(P^*)^{\mathrm{HB}}$ which outputs accepting proofs with probability at least $p - m\varepsilon(n)$, by defining $(P^*)^{\mathrm{HB}}(x, \sigma) = P^*(x, c)$ where $(c_1, \ldots, c_m) = (\mathrm{Sen}(x, \sigma_1), \ldots, \mathrm{Sen}(x, \sigma_m))$. Since $(P^*)^{\mathrm{HB}}$ can produce an accepting transcript with only negligible probability, P^* produces an accepting proof with negligible probability. Therefore, the soundness of $(P^{\mathrm{HB}}, V^{\mathrm{HB}})$ carries over to (D, P, V).

3. *Zero Knowledge.* We construct the following simulator S for the proof system. We let S be a pair of PPTs (S^{HB}, S'), where S^{HB} is the simulator for the hidden bits NIZK proof system for Π. S^{HB} takes in as input $x \in \Pi$, and outputs (σ_I, I, π). S' takes in σ_I as input, randomly completes σ by selecting the bits not in σ_I, and generates commitment/opening pairs (c_i, o_i) for all bits σ_i (the pairs are drawn randomly from the possible choices of commitments and openings).

   In order to show that S can truly simulate real transcripts, we first build the following distributions:

   - The distributions of real transcripts, generated by the dealer D and the prover P:
     $H_0 = \{c \leftarrow D(1^k), (I, \sigma^P, o^P, \pi) \leftarrow P(x, c) : (c, \sigma_I^P, I, o_I^P, \pi)\}$
   - A hybrid for which a modified dealer D' not only sends c, but also the openings o to the prover P^{HB}.
     $H_1 = \{(\sigma^D, c, o^D) \leftarrow D'(1^k), (I, \pi) \leftarrow P^{\text{HB}}(x, \sigma^D) : (c, \sigma_I^D, I, o_I^D, \pi)\}$
   - A hybrid where σ is generated uniformly and fed to P^{HB} to produce (I, π), as well as to a modified dealer D'', which on input σ, x produces the pair (c, o) for σ.
     $H_2 = \{\sigma \leftarrow \{0,1\}^m, (I, \pi) \leftarrow P^{\text{HB}}(x, \sigma), (c, o) \leftarrow D''(\sigma, x) : (c, \sigma_I, I, o_I, \pi)\}$
   - The distribution of simulated transcripts:
     $H_3 = \{(\sigma_I, I, \pi) \leftarrow S^{\text{HB}}, (\sigma \backslash \sigma_I, c, o) \leftarrow S'(\sigma_I, I) : (c, \sigma_I, I, o_I, \pi)\}$
     where by $\sigma \backslash \sigma_I$ we refer to those bits of σ which had not already been selected by the choice of σ_I.

   We now proceed to prove the indistinguishability relationships between these different hybrids. By examination, we see that H_0, H_1 and H_2 are identically distributed. By the properties of hidden bits zero knowledge proof systems, we know that the transcripts produced by P^{HB}, $\{\sigma \leftarrow \{0,1\}^m, (I, \pi) \leftarrow P^{\text{HB}}(x, c) : (\sigma_I, I, \pi)\}$ are statistically indistinguishable from those simulated by S^{HB}, $\{(\sigma_I, I, \pi) \leftarrow S^{\text{HB}} : (\sigma_I, I, \pi)\}$, so the σ_I, I, π fragments of the hybrids H_1 and H_2 are statistically indistinguishable. In both cases, the commitments c_I and openings o_I to the bits in σ_I are generated using the sender algorithm Sen, so the distributions remain statistically indistinguishable if we include these. The distributions differ, however, in how the other commitments $c \backslash c_I$ are generated. In H_2, these are commitments to bits $\sigma \backslash \sigma_I$ that are correlated with (σ_I, I, π). In H_3, they are commitments to bits $\sigma \backslash \sigma_I$ that are uniform and independent of (σ_I, I, π). But, by the hiding property, commitments to any two sequences of bits are computationally indistinguishable. Hence H_0 and H_3, representing the real and, respectively, the simulated transcripts, are computationally indistinguishable, proving that the proof system (D, P, V) is zero knowledge.

   If the commitment scheme is statistically rather than computationally hiding on NO instances, then the ensembles above are statistically indistinguishable, and we obtain a NISZK^{h} proof system.

■

Remarks. We make the following observations about the protocol in the proof of Theorem 5.9.

1. If the commitment scheme is not instance-dependent, but rather depends only on the security parameter (i.e., the length of the input x), then we obtain a proof system in the *public parameter* model. Combining this with the construction of commitments from one-way functions [HILL, Nao], we get another proof of the fact that one-way functions imply $\mathrm{NIZK}^{\mathrm{pub}} = \mathrm{AM}$ [BG, Pas].

2. The protocol requires a computationally unbounded honest prover, because the prover must break the commitments. However, the prover can be implemented efficiently in a generalization of the help model where the dealer can generate secret information (e.g. the openings to the commitments) for the prover in addition to the common reference string. Such a model can be useful for applications of noninteractive zero knowledge where the dealer and the honest prover are the same party, such as the Bellare–Goldwasser signature scheme [BG]. (This signature scheme also requires that the zero knowledge property holds even when *many, adaptively chosen* statements are proven using the same reference string; unfortunately, our construction does not provide such guarantees.) This model for noninteractive zero knowledge should be contrasted with one where the *verifier* receives secret information from the dealer, which has proven useful in the construction of encryption schemes secure against chosen-ciphertext attack [CS], and one where both parties receive secret information, as studied in [CD].

6 From ZK^{h} to ZK

We generalize the results of Ben-Or and Gutfreund [BG] that $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$ (Theorem 3.3) to show that adding help to ZK proofs does not confer any additional power:

Theorem 6.1 ***(Theorem 1.3, restated)*** $\mathrm{ZK}^{\mathrm{h}} = \mathrm{ZK}$.

To prove Theorem 3.3, Ben-Or and Gutfreund employ the techniques of [AH, PT, GV], by considering the output of the simulator S for a zero-knowledge proof for Π as the moves of a *virtual prover* and a *virtual verifier.* The simulated transcripts are compared to the transcripts output by a cheating strategy for a real prover P_S (called the *simulation-based prover*), which tries to imitate the behavior of the virtual prover. Intuitively, on YES instances, the output of the simulator should be statistically close to the output of the simulation-based prover interacting with the real verifier. On NO instances, however, if we modify the simulator to accept with high probability (we can easily modify it to do that), the difference between the two transcripts must be significant. [BG] exploit this to show that any problem in $\mathrm{SZK}^{\mathrm{h}}$ can be reduced to the intersection of the SZK-complete problems Statistical Difference([SV]) and Entropy Difference([GV]). Since the other direction ($\mathrm{SZK} \subseteq \mathrm{SZK}^{\mathrm{h}}$) follows from the definitions, the conclusion that $\mathrm{SZK} = \mathrm{SZK}^{\mathrm{h}}$ follows immediately. We will use the same strategy with ZK^{h}, replacing statistical measures of closeness with computational ones. To do this, we will replace the SZK-complete problems SD and ED the Indistinguishability Condition and the Conditional Pseudoentropy Condition, which characterize the class ZK (Theorems 3.7 and 3.9).

We will use the following notation throughout this section: we let (D, P, V) be a ZK proof system for promise problem Π, and we let S be the simulator for the honest verifier V. We assume that the verifier uses a total of $r = r(|x|)$ coins. Including the dealer's message, we assume that $2l$

messages make up a transcript, where $l = l(|x|)$, and that each message has length r. Additionally, the last message reveals the verifier's random coins. We use the notation $S(x)$ to refer to the simulated transcripts. For a transcript γ, we denote γ_i the prefix of γ consisting of the first i messages.

We construct the simulation-based prover in the following manner: for an odd i, given a conversation prefix $\gamma \in \{0,1\}^{(i-1)r}$, the next message of P_S is:

1. If the probability that $S(x)$ outputs a conversation with prefix γ is 0, then P_S sends a dummy message, say 0^r.
2. Otherwise, P_S replies with the same conditional probability as the virtual prover, sending β with probability $\Pr[S(x)_i = \gamma\beta | S(x)_{i-1} = \gamma]$.

Note that P_S sends the first message instead of the dealer, using the simulator to generate the help string. Define $\langle P_S, V\rangle(x)$ to be the distribution of the possible transcripts of conversations between P_S and V.

Lemma 6.2 *[AH, PT, GV, BG] For all x, $\mathrm{KL}(S(x)|\langle P_S, V\rangle(x)) = r - \sum_{i=1}^{l}[\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})]$.*

Lemma 6.3 *[AH, PT, GV, BG] For $x \in \Pi_N$, let p denote the probability that $S(x)$ outputs an accepting transcript. Suppose that $\Delta(D(x), S(x)_1) \le q_1$. Denote by $q_2 = q_2(|x|)$ the soundness of the protocol. Let $q = 2q_1 + q_2$, and suppose that $p \ge q$. Then,*

$$\mathrm{KL}(S(x)|\langle P_S, V\rangle(x)) \ge \mathrm{KL}_2(p, q).$$

We will use the previous two lemmas to prove the main result of this section:

Proof of Theorem 6.1 Since $\mathrm{ZK} \subseteq \mathrm{ZK^h}$ by definition, we prove $\mathrm{ZK^h} \subseteq \mathrm{ZK}$. Consider a problem Π with a $\mathrm{ZK^h}$ proof system with completeness and soundness errors at most $(2lr)^{-2}/2$. We modify the proof system such that 0^{2lr} is always an accepting transcript, and such that the simulator always outputs accepting transcripts (e.g., swap on rejecting transcripts with 0^{2lr}). The new proof system has soundness error at most $2^{-r} + (2lr)^{-2}/2$.

Similarly to [BG, GV, Vad], consider the following distributions:

- $X_{x,1} = (S(x)_2, \ldots S(x)_{2l})$, $Y_{1,x} = (S(x)_1, \ldots S(x)_{2l-1})$.
- $X_{x,2} = D(x)$, $Y_{2,x} = S(x)_1$.

Claim 6.4 *If $x \in \Pi$, $X_{2,x} \overset{c}{\equiv} Y_{2,x}$ and $(X_{1,x}, Y_{1,x}) \overset{c}{\equiv} (X', Y')$, where $\mathrm{H}(X'|Y') = r$.*

Proof: When $x \in \Pi_Y$, $X_{2,x} \overset{c}{\equiv} Y_{2,x}$ and $(X_{1,x}, Y_{1,x}) \overset{c}{\equiv} (X', Y')$, where (X', Y') is the distribution of real transcripts produced by $\langle D, P, V\rangle$. That is, $X' = (\langle D, P, V\rangle(x)_2, \ldots \langle D, P, V\rangle(x)_{2l})$ and $Y' = (\langle D, P, V\rangle(x)_1, \ldots \langle D, P, V\rangle(x)_{2l-1})$.

The conditional entropy of X' given Y' will be:

$$\mathrm{H}(X'|Y') = \sum_{i=1}^{l} \mathrm{H}(\langle D, P, V\rangle(x)_{2i} | \langle D, P, V\rangle(x)_{2i-1}) = r$$

since the sum measures the total entropy contributed by the verifier's messages. ■

Claim 6.5 *If $x \in \Pi$, either $\Delta(X_{2,x}, Y_{2,x}) \geq (2lr)^{-1}$ or $\mathrm{H}(X_{1,x}|Y_{1,x}) \leq r - 1$.*

Proof: Assume $\Delta(X_{2,x}, Y_{2,x}) \leq (2lr)^{-1}$. Then, we have:
or, by Lemmas 6.2 and 6.3 and assuming that $2^{-r} + (2lr)^{-2}/2 + (lr)^{-1} < 1/2, q_1 = (2lr)^{-1}, q_2 = 2^{-l} + (2lr)^{-2}/2$, and $p = 1$:

$$\begin{aligned}
&\mathrm{H}(X_{1,x}|Y_{1,x}) \\
&= \sum_{i=1}^{l} \mathrm{H}(S(x)_{2i}|S(x)_{2i-1}) \\
&= \sum_{i=1}^{l} \mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})) \\
&= r - \mathrm{KL}(S(x)|\langle P_S, V\rangle(x)) \text{ (by Lemma 6.2)} \\
&\leq r - \mathrm{KL}_2(1, 1/2) \text{ (by Lemma 6.3, with } p = 1, q_1 = (2lr)^{-1}, q_2 = 2^{-l} + (2lr)^{-2}/2, q = q_1 + 2q_2 \leq 1/2) \\
&= r - \log 2 \\
&= r - 1
\end{aligned}$$

■

Having mapped instances $x \in \Pi$ to $(X_{x,1}, Y_{x,1})$ and $(X_{x,2}, Y_{x,2})$, consider the promise problems Γ and Λ defined by $\Gamma_Y = \Lambda_Y = \Pi_Y, \Gamma_N = \{x \in \Pi_N : \Delta(X_{x,2}, Y_{x,2}) \geq (2lr)^{-1}\}$ and $\Lambda_N = \{x \in \Pi_N : \mathrm{H}(X_{x,1}, Y_{x,1}) \leq r - 1\}$. Then $\Pi = \Gamma \cap \Lambda$ (i.e., $\Pi_Y = \Gamma_Y \cap \Lambda_Y$ and $\Pi_N = \Gamma_N \cup \Lambda_N$). Since ZK is closed under intersection (run protocols for Γ and Λ in parallel), it suffices to show that both $\Gamma \in$ ZK and $\Lambda \in$ ZK. Both Γ and Λ are in IP; this follows because they are restrictions of Π, which is in $\mathrm{ZK}^{\mathrm{h}} \subseteq$ IP. Γ satisfies the Indistinguishability Condition (the inverse polynomial statistical difference can be amplified to 2/3 by taking direct products), so $\Gamma \in$ ZK (by Theorem 3.7), and Λ satisfies the Conditional Pseudoentropy Condition, so $\Lambda \in$ ZK (by Theorem 3.9). Consequently $\Pi \in \mathrm{ZK} \subseteq \mathrm{IP}$.

7 Putting it all together

Theorem 7.1 ***(Theorem 1.2, restated)*** $\mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK} = \mathrm{NISZK}^{\mathrm{h}}$.

Proof: Theorem 3.3 implies $\mathrm{NISZK}^{\mathrm{h}} \subseteq \mathrm{SZK}^{\mathrm{h}} = \mathrm{SZK}$. To prove $\mathrm{SZK} \subseteq \mathrm{NISZK}^{\mathrm{h}}$, recall that any problem in SZK has an instance-dependent commitment scheme that is statistically hiding on YES instances and statistically binding on NO instances by Theorem 5.4. By Lemma 5.8, the scheme can be made noninteractive and statistically binding for honest senders. Since SZK ⊆ AM [AH], the construction of Theorem 5.9 can be applied to yield a $\mathrm{NISZK}^{\mathrm{h}}$ proof. ■

Theorem 7.2 ***(Theorem 1.1, restated)*** $\mathrm{ZK}^{\mathrm{h}} \cap \mathrm{AM} = \mathrm{ZK} \cap \mathrm{AM} = \mathrm{NISZK}^{\mathrm{h}}$.

Proof: The proof is analogous to the statistical zero knowledge case. ■

8 Other characterizations

In this section, we obtain new characterizations of ZK and SZK, by using their relationships with $\mathrm{NIZK}^{\mathrm{h}}$ and $\mathrm{NISZK}^{\mathrm{h}}$, established in the previous sections. We start by considering a variant of the $\mathrm{NISZK}^{\mathrm{h}}$-complete problem given by Ben-Or and Gutfreund [BG]:

Definition 8.1 Image Intersection Density (IID)

$$\begin{aligned} \mathrm{IID}_Y &= \{(X,Y) : \Delta(X,Y) < 1/3\} \\ \mathrm{IID}_N &= \{(X,Y) : (X,Y) \textit{ are mutually } 2/3\textit{-disjoint}\} \end{aligned}$$

In [BG], the thresholds of 1/3 and 2/3 are replaced with $1/\mathrm{poly}(n)$ and $1 - 1/\mathrm{poly}(n)$; we will prove that the above version is $\mathrm{NISZK}^{\mathrm{h}}$-complete by using the following strengthening of the Polarization Lemma given in [BG]:

Lemma 8.2 (***Polarization Lemma, based on [BG, SV]***) *For all constants such that $0 \leq \alpha < \beta \leq 1$, there exists a polynomial-time procedure that takes a pair of distributions (X_0, X_1) and a parameter n in unary, and outputs a pair of distributions (Y_0, Y_1) such that:*

1. $\Delta(X_0, X_1) < \alpha \Rightarrow \Delta(X_0, X_1) < 2^{-n}$.

2. (X_0, X_1) *is mutually* β*-disjoint* $\Rightarrow (Y_0, Y_1)$ *is mutually* $(1 - 2^{-n})$*-disjoint.*

The proof of the Polarization Lemma can be found in Appendix A.

Our Polarization Lemma is stronger than the one stated in [BG], which only achieves polarization from thresholds $\alpha = 1/\mathrm{poly}(n), \beta = 1 - 1/\mathrm{poly}(n)$. Using a combination of the tools in [BG] and [SV], we achieve the constant threshold polarization we present.

This can be compared to the original Polarization Lemma of [SV], which refers to statistical difference in Item 2 (rather than mutual disjointness), but only achieves polarization from thresholds such that $0 \leq \alpha < \beta^2 \leq 1$, and for which there is evidence that the gap between thresholds is inherent ([HR]).

We also add that, at a factor of 2 in β, we can start with β-disjoint distributions rather than mutually β-disjoint ones for the polarization to work. The reason is that we can easily transform a pair (X, Y) that is 2β-disjoint into a pair (X', Y') such that $\Delta(X', Y') = \Delta(X, Y)$ and (X', Y') is mutually β-disjoint, using Lemma B.1.

We also give a computational analogue of IID:

Definition 8.3 (CIIDC) *A promise problem* Π *satisfies the* Computational Image Intersection Density Condition *(*CIIDC*) if there is a polynomial time mapping from strings $x \in \Pi$ to two efficiently samplable distributions (X, Y) and a parameter $m = \mathrm{poly}(|x|)$ such that*

1. *If $x \in \Pi_Y$, then X and Y are computationally indistinguishable.*

2. *If $x \in \Pi_N$, then (X, Y) are mutually* $1/3$*-disjoint.*

We observe that IID and CIIDC exactly capture noninteractive, instance-dependent commitments that are statistically binding for an honest sender:

Lemma 8.4 *A promise problem* Π *has a noninteractive, instance-dependent commitment scheme that is statistically (resp., computationally) hiding on* YES *instances and statistically binding for honest senders on* NO *instances if and only if* Π *reduces to* IID *(resp., if and only if* Π *satisfies the* CIIDC*).*

Proof: For the backwards direction, consider a problem Π which reduces to IID (the computational case will be similar). We construct the following protocol:

Commitment protocol for Π:

1. **Preprocessing:**

   First, reduce $x \in \Pi$ to an instance (X_0, X_1) of IID. Use the Polarization Lemma on (X_0, X_1) to obtain (Y_0, Y_1) such that, if $x \in \Pi_Y$, $\Delta(Y_0, Y_1) \leq 2^{-n}$, and, if $x \in \Pi_N$, (Y_0, Y_1) are mutually $(1 - 2^{-n})$-disjoint, where $n = |x|$.

2. **Commit Stage:**

   $S_x(x, b)$: To commit to bit $b \in \{0, 1\}$, choose $d \overset{R}{\leftarrow} \{0,1\}^m$, where m is the input length of Y_b, set $c = Y_b(d)$ and output (c, d).

3. **Reveal Stage:**

   $R_x(x, c, b, d)$: Accept if and only if $Y_b(d) = c$.

On $x \in \Pi_Y$, we know that Y_0 and Y_1 have negligible statistical difference. Hence, a commitment to 1 is statistically indistinguishable from a commitment to 0. Hence, the scheme is computationally hiding on YES instances (actually, the scheme is statistically hiding.)

When $x \in \Pi_N$, the pair (Y_0, Y_1) is mutually $(1 - 2^{-n})$-disjoint. It directly follows that only a negligible fraction of commitments can be opened in two ways.

In the case that we are working with a problem which satisfies the CIIDC, we use the same scheme. However, instead of polarizing, we will simply take direct products to amplify the mutual disjointness on NO instances while preserving computational indistinguishability on YES instances (Lemma A.1).

For the forwards direction, let $\text{Com}_x = (S_x, R_x)$ be a noninteractive, instance-dependent commitment scheme that is statistically hiding on YES instances and statistically binding for honest senders on NO instances, and consider $X = S_x(0)$ and $Y = S_x(1)$.

- If $x \in \Pi_Y$, we know that $\Delta(\text{view}_R(S_x(0), R), \text{view}_R(S_x(1), R)) \leq \varepsilon(|x|)$, and hence, $\Delta(S_x(0), S_x(1)) \leq \varepsilon(|x|)$.
- If $x \in \Pi_N$, assume that there exists no negligible function $\mu(|x|)$ such that $(S_x(0), S_x(1))$ are mutually $(1 - \mu(|x|))$-disjoint. Hence for all negligible functions $\mu(|x|)$ and $c \leftarrow S_x(b)$, $\Pr\left[c \in S_x(\bar{b})\right] > \mu(|x|)$. But then, S can always succeed with probability greater than $\mu(|x|)$ at the game described in Definition 5.7. So, for some negligible μ, $(S_x(0), S_x(1))$ is mutually $(1 - \mu(|x|))$-disjoint, and Π reduces to IID.

The proof for the computational case is analogous. ■

Theorem 8.5 *The following hold:*

1. IID *is complete for* SZK = $\mathrm{SZK}^{\mathrm{h}}$*.*

2. $\Pi \in \mathrm{ZK} = \mathrm{ZK}^{\mathrm{h}}$ *if and only if* $\Pi \in \mathrm{IP}$ *and* Π *satisfies the* CIIDC*.*

Proof: We prove Item 2: Since a promise problem which satisfies CIIDC also satisfies the Indistinguishability Condition (this follows from the fact that of two distributions are α-disjoint, they must have statistical difference at least α), the promise problem must have a ZK proof system by Theorem 3.7. Conversely, any problem in $\mathrm{ZK}^{\mathrm{h}} = \mathrm{ZK}$ has a instance-dependent commitment scheme that is computationally hiding on YES instances and statistically binding on NO instances. By Theorem 5.8, this can be transformed into a noninteractive, instance-dependent commitment scheme that is computationally hiding on YES instances and statistically binding for honest senders on NO instances and thus satisfies CIIDC.

The proof of Item 1 is analogous. ■

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.

[BG] M. Bellare and S. Goldwasser. New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero Knowledge Proofs. In *CRYPTO '89*, pages 194–211, 1989.

[BMO] M. Bellare, S. Micali, and R. Ostrovsky. Perfect zero-knowledge in constant rounds. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 482–493, 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 *CRYPTO '88*, pages 37–56, 1988.

[BG] M. Ben-Or and D. Gutfreund. Trading Help for Interaction in Statistical Zero-Knowledge Proofs. *Journal of Cryptology*, 16(2), March 2003. Preliminary version appeared as [GB].

[BDMP] M. Blum, A. De Santis, S. Micali, and G. Persiano. Noninteractive Zero-Knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, Dec. 1991.

[BFM] M. Blum, P. Feldman, and S. Micali. Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract). In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112, 1988.

[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.

[Cio] D. Ciocan. Constructions and Characterizations of Non-Interactive Zero-Knowledge. Undergradute thesis, Harvard University, 2007.

[CD] R. Cramer and I. Damgaard. Secret-Key Zero-Knowledge and Non-Interactive Verifiable Exponentiation. In *ACR Theory of Cryptography Conference (TCC '04)*, pages 223–237. Springer-Verlag, 2004.

[CS] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *EUROCRYPT '02: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pages 45–64, London, UK, 2002. Springer-Verlag.

[FLS] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero Knowledge Proofs Under General Assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[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.

[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, 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 *CRYPTO '99*, pages 467–484, 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.

[GB] D. Gutfreund and M. Ben-Or. Increasing the Power of the Dealer in Non-interactive Zero-Knowledge Proof Systems. In *ASIACRYPT '00: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security*, pages 429–442, London, UK, 2000. Springer-Verlag. Journal version appeared as [BG].

[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.

[HR] T. Holenstein and R. Renner. One-Way Secret-Key Agreement and Applications to Circuit Polarization and Immunization of Public-Key Encryption. In *Advances in Cryptology CRYPTO 2005*, pages 478–493, New York, NY, USA, 2005. ACM Press.

[IY] R. Impagliazzo and M. Yung. Direct Minimum-Knowledge Computations (Extended Abstract). In *CRYPTO '87: A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, pages 40–51, London, UK, 1988. Springer-Verlag.

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

[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.

[MV] D. Micciancio and S. Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In *CRYPTO*, pages 282–298, 2003.

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

[NV] M.-H. Nguyen and S. Vadhan. Zero knowledge with efficient provers. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 287–295, New York, NY, USA, 2006. ACM Press.

[Ong] S. J. Ong. *Unconditional Relationships within Zero Knowledge.* PhD thesis, Harvard School of Engineering and Applied Sciences, 2005.

[OV1] S. J. Ong and S. Vadhan. An Equivalence between Zero Knowledge and Commitments, 2007. In submission. Results can be found in [Ong].

[OV2] S. J. Ong and S. Vadhan. Zero Knowledge and Soundness are Symmetric. In *EUROCRYPT '07: 26th Annual Conference on the Theory and Applications of Cryptographic Techniques*, 2007.

[Pas] R. Pass and abhi shelat. Unconditional Characterizations of Non-Interactive Zero-Knowledge. In *CRYPTO '05*, pages 118–134. Springer Berlin / Heidelberg, 2005.

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

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

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

[Vad] S. Vadhan. An Unconditional Study of Computational Zero Knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. Special Issue on Randomness and Complexity.

A Proof of the Polarization Lemma (Lemma 8.2)

In order to prove our version of the Polarization Lemma, we give a few helper lemmas.

Lemma A.1 ***[BG, SV]*** *Given a pair of distributions (X_0, X_1) with n input gates and a parameter k, consider the following distributions:*

Y_0*: Choose* $(r_1, \ldots, r_k) \overset{R}{\leftarrow} \{0,1\}^{kn}$*, output* $(X_0(r_1), \ldots, X_0(r_k))$*.*

Y_1*: Choose* $(r_1, \ldots, r_k) \overset{R}{\leftarrow} \{0,1\}^{kn}$*, output* $(X_1(r_1), \ldots, X_1(r_k))$*.*

The following properties hold:

1. $\Delta(Y_0, Y_1) \leq k \cdot \Delta(X_0, X_1)$.

2. If the pair (X_0, X_1) *is mutually* α*-disjoint, then* (Y_0, Y_1) *is mutually* $(1-(1-\alpha)^k)$*-disjoint.*

Lemma A.2 ***[BG, SV]*** *Given two pairs* (X_0, X_1) *and* $(X_0^{'}, X_1^{'})$*, with* n *and* $n^{'}$ *input gates, respectively, consider the circuits:*
Y_0*: Choose* $b \overset{R}{\leftarrow} \{0,1\}, r \overset{R}{\leftarrow} \{0,1\}^n, r^{'} \overset{R}{\leftarrow} \{0,1\}^{n^{'}}$*, output* $(X_b(r), X_b^{'}(r^{'}))$.
Y_1*: Choose* $b \overset{R}{\leftarrow} \{0,1\}, r \overset{R}{\leftarrow} \{0,1\}^n, r^{'} \overset{R}{\leftarrow} \{0,1\}^{n^{'}}$*, output* $(X_b(r), X_{\overline{b}}^{'}(r^{'}))$.
The following properties hold:

1. $\Delta(Y_0, Y_1) = \Delta(X_0, X_1) \cdot \Delta(X_0^{'}, X_1^{'})$.

2. If (X_0, X_1) *and* $(X_0^{'}, X_1^{'})$ *are mutually* α*-disjoint and* $\alpha^{'}$ *respectively, then* (Y_0, Y_1) *are mutually* $(\alpha\alpha^{'})$*-disjoint.*

Lemma A.3 ***[BG, SV]*** *Given circuits* X_0, X_1 *with* n *input gates and a parameter* k*, consider the following pair:*
Y_0*: Choose* $(b_1, \ldots, b_k) \overset{R}{\leftarrow} \{(c_1, \ldots, c_k) \in \{0,1\}^k : c_1 \oplus \ldots \oplus c_k = 0\}, (r_1, \ldots r_k) \overset{R}{\leftarrow} \{0,1\}^{kn}$*, output* $(X_{b_1}(r_1), \ldots, X_{b_k}(r_k))$.
Y_1*: Choose* $(b_1, \ldots, b_k) \overset{R}{\leftarrow} \{(c_1, \ldots, c_k) \in \{0,1\}^k : c_1 \oplus \ldots \oplus c_k = 1\}, (r_1, \ldots r_k) \overset{R}{\leftarrow} \{0,1\}^{kn}$*, output* $(X_{b_1}(r_1), \ldots, X_{b_k}(r_k))$.
The following properties hold:

1. $\Delta(Y_0, Y_1) = \Delta(X_0, X_1)^k$.

2. If the pair (X_0, X_1) *is mutually* α*-disjoint, then* (Y_0, Y_1) *is mutually* α^k*-disjoint.*

Using the above lemmas, we are ready to prove our version of the Polarization Lemma:

Lemma A.4 ***[BG, SV] (Polarization Lemma)*** *There exists a polynomial-time procedure that takes a pair of distributions* (X_0, X_1) *with description length* n*, and outputs a pair of distributions* (Y_0, Y_1) *such that:*

1. $\Delta(X_0, X_1) < \alpha \Rightarrow \Delta(X_0, X_1) < 2^{-n}$.

2. (X_0, X_1) *is mutually* β*-disjoint* $\Rightarrow (Y_0, Y_1)$ *is mutually* $(1-2^{-n})$*-disjoint,*

for all constants α *and* β *such that* $\alpha < \beta$.

Proof: Let $\lambda = \min\{\beta/\alpha, 2\} > 1$.

We first apply Lemma A.3 with $k = \log_\lambda 2n$, obtaining two distributions which are either α^k statistically close, or mutually $(\beta)^k$-disjoint.

Then, apply Lemma A.1 with $m = \frac{\lambda^k}{2\beta^k} \leq \frac{1}{2\alpha^k}$. This gives 2 distributions with either:

- Statistical difference at most $m\alpha^k \leq 1/2$.
- Mutual disjointness of at most $1-(1-\beta^k)^m \geq 1 - e^{-\beta^k m} = 1 - e^{-\beta^k \frac{\lambda^k}{2\beta^k}} = 1 - e^{-\frac{\lambda^k}{2}} = 1 - e^{-n}$.

Finally, we again apply Lemma A.3 with parameter n to get either statistical difference at most 2^{-n}, or mutual disjointness at most $(1-e^{-n})^n \geq 1 - ne^{-n} \geq 1 - 2^{-n}$. ■

B From Disjoint to Mutually-Disjoint Distributions

Lemma B.1 ***[BG, SV]*** *Given a pair of distributions* (X_0, X_1) *with* n *input gates, consider the following distributions:*

Y_0*: Choose* $r \overset{R}{\leftarrow} \{0,1\}^n, b \overset{R}{\leftarrow} \{0,1\}$*, output* $(X_b(r), b)$*.*

Y_1*: Choose* $r \overset{R}{\leftarrow} \{0,1\}^n, b \overset{R}{\leftarrow} \{0,1\}$*, output* $(X_b(r), \overline{b})$*.*

The following properties hold:

1. $\Delta(Y_0, Y_1) = \Delta(X_0, X_1)$

2. *If* (X_0, X_1) *is* α*-disjoint, then* (Y_0, Y_1) *is mutually* $\frac{\alpha}{2}$*-disjoint.*