J. Cryptol. (2015) 28: 533–550
DOI: 10.1007/s00145-013-9165-6

Journal of CRYPTOLOGY

How to Achieve Perfect Simulation and a Complete Problem for Non-interactive Perfect Zero-Knowledge

Lior Malka
Department of Computer Science, University of Victoria, Victoria, BC, Canada
lior34@gmail.com

Communicated by Tatsuaki Okamoto

Received 12 October 2009
Online publication 21 November 2013

Abstract. This paper studies *perfect* zero-knowledge proofs. Such proofs do not allow any simulation errors, and therefore techniques from the study of *statistical* zero-knowledge (where a small error is allowed) do not apply to them. We introduce a new *error shifting technique* for building perfect simulators. Using this technique we give the first complete problem for the class of problems admitting non-interactive perfect zero-knowledge (**NIPZK**) proofs, a hard problem for the class of problems admitting public-coin **PZK** proofs, and other applications.

Key words. Cryptography, Non-interactive, Perfect zero-knowledge, Perfect simulation, Error shifting, Complete problems.

1. Introduction

Perfect zero-knowledge protocols allow one party (the *prover*) to prove the validity of an assertion to another party (the *verifier*), but without leaking any information [14]. This is formalized using the notion of a simulator, and requiring that the simulation error be zero. The notion of perfect zero knowledge can be relaxed to *statistical* zero knowledge, where the prover leaks a negligible amount of information, and *computational* zero knowledge, where this leakage is not noticeable by computationally bounded verifiers.

The past few years have seen great progress in proving general results about the class of problems admitting statistical zero-knowledge (**SZK**) proofs. These results provide complete problems and show equivalence between private and public-coin proofs, honest and malicious verifiers, efficient provers, and more ([10,12,21,22,26,32]). Various techniques, such as lower-bound protocols [13] and transformations that polarize and reverse the statistical distance represented by circuits [26], were used in proving these results. Unfortunately, these and other techniques used in the study of statistical zero-knowledge proofs do not apply to the class of problems admitting perfect zero-knowledge (**PZK**) proofs. Intuitively, these techniques manipulate the protocol in a way

that introduces a small error into the simulation. This is not an issue in the case of statistical zero knowledge, where a small simulation error is allowed, but it is an issue in the case of perfect zero knowledge, where no simulation error is allowed. Consequently, many fundamental questions about **PZK** remain open.

Perfect zero-knowledge protocols are interesting from a cryptographic perspective because they provide the maximum level of privacy for the prover. Under certain complexity assumptions, every language in **NP** has a perfect zero-knowledge argument [4,20], and recently a non-interactive argument was discovered [15] (an *argument* is a computationally sound proof). Their simple definition also makes them ideal to use as a testbed for studying zero knowledge in new settings. Recent examples include the local zero-knowledge protocol of [19] and the quantum zero-knowledge protocol of [33]. From a complexity-theoretic perspective, there are well-known problems that unconditionally admit **PZK** proofs, such as QUADRATIC-RESIDUOUSITY, DISCRETE-LOG, and GRAPH-ISOMORPHISM [9,14,31]. These problems are in **NP**, but not known to be in **P** or **NP**-complete. Moreover, they all admit 3-round proofs, yet we do not know whether **PZK** proofs can be made to have a constant number of rounds (this was recently proven for **SZK** [23], but the techniques do not extend to **PZK**). Our goal is to develop tools that will facilitate the study of perfect zero-knowledge proofs.

1.1. *Our Results*

As was mentioned earlier, techniques used in the study of statistical zero-knowledge proofs introduce error into the simulation, and therefore cannot be applied to perfect zero-knowledge proofs. To overcome this difficulty we introduce what we call *an error shifting technique*. Roughly speaking, the idea is to first identify where the error is coming from, and then shift it forward to the protocol in a way that does not affect the simulation (but may affect the completeness or soundness errors). This is in contrast to techniques from the statistical setting, where the error is incorporated into the constructions, thus leading to simulation errors later on. Since the notion of simulation is central to cryptography, our technique may be useful for achieving perfect simulation in contexts outside of zero-knowledge.

The first domain to which we apply the error shifting technique is complete problems. Recall that a problem Π is said to be *hard* for some complexity class $\mathcal{C}$ if every problem in the class $\mathcal{C}$ efficiently reduces to it. The problem Π is said to be *complete* for $\mathcal{C}$ if Π is hard for $\mathcal{C}$ and Π is in $\mathcal{C}$. Complete problems are a powerful tool because they represent an entire class. Thus, by proving a result with respect to a complete problem we get a general result about the entire class. Indeed, most of the study of statistical zero-knowledge proofs was made possible by first finding complete problems and then using them to prove more advanced results. This also means that providing complete problems for the perfect setting is an important step towards translating the results from the statistical setting to the perfect setting.

We obtain complete and hard problems in both the interactive and the non-interactive setting. In the non-interactive setting we consider STATISTICAL DISTANCE FROM UNIFORM (SDU), the complete problem of Goldreich Sahai and Vadhan [11] (based on [30]) for the class of problems admitting non-interactive statistical zero-knowledge (**NISZK**) proofs. Instances of this problem are circuits that represent distributions, namely the output distribution of the circuit when the input to the circuit is uniformly

distributed. YES instances are circuits that represent distributions that are statistically close to uniform, and NO instances have a small support.

Here, we obtain the first complete problem for the class of problems admitting non-interactive perfect zero-knowledge (**NIPZK**) proofs:

Theorem 1. *The problem* UNIFORM (UN) *is* **NIPZK***-complete.*

Our problem UNIFORM is similar to SDU, except that YES instances of UNIFORM are circuits *exactly* representing the uniform distribution rather than merely being statistically close, but where the circuits also have an additional output bit used for shifting the error forward. Intuitively, we shift the error from the reduction, through the circuits, and into the protocol. The difference between UNIFORM and SDU is natural as it reflects the difference between perfect and statistical simulation.

Turning our attention to the interactive model, we consider STATISTICAL DISTANCE (SD), the complete problem of Sahai and Vadhan [26] for the class of problems admitting statistical zero-knowledge (**SZK**) proofs. Instances of this problem are pairs $\langle X, Y \rangle$ of circuits. As YES instances, X and Y represent statistically close distributions, and as NO instances, X and Y are represent statistically far distributions. In the case of public-coin **HVPZK** problems with perfect completeness, [26] showed that a similar reduction yields circuits X and Y that are identically distributed as YES instances. Using the error shifting technique, we remove the restriction on perfect completeness and obtain the problem IDENTICAL DISTRIBUTIONS. This problem is similar to the perfect variant of SD, except that it introduces a third circuit to the instance.

Theorem 2. *The problem* IDENTICAL DISTRIBUTIONS *is hard for the class of problems admitting* **public-coin-PZK** (*and even* **public-coin-HVPZK**) *proofs.*

Our theorems and the error shifting technique can facilitate the study of perfect zero-knowledge proofs in both the interactive and the non-interactive setting. For example, our hard problem was used in [18] to study the round complexity of perfect zero-knowledge proofs and to prove an equivalence between zero knowledge and instance-dependent commitment schemes in the perfect setting (a more meaningful equivalence was recently given [23], but it only applies to the statistical and the computational settings). We give two additional applications.

The first application shows equivalence between the notion of zero knowledge where the simulator is allowed to fail (also known as *abort*) to the notion of zero knowledge where the simulator is not allowed to fail. This result is with respect to the honest verifier (more accurately, any fixed verifier). The second application considers closure properties of **NIPZK**. That is, using UNIFORM, we give **NIPZK** proofs for the OR of any two **NIPZK** problems admitting very small completeness and soundness errors. We mention that no such closure result is known in the case of non-interactive statistical zero-knowledge (**NISZK**) proofs.

1.2. *Related Work*

To the best of our knowledge, the only general result about perfect zero-knowledge proofs is due to [5], who showed a transformation from honest-verifier **PZK** proofs

to malicious-verifier **PZK** proofs. This transformation applies only to constant-round, public-coin proofs.

Our work is inspired by the study of statistical zero-knowledge proofs, and we build on the results of [11,26] (based on [1,7,22,30]). Sahai and Vadhan [26] showed a **HVPZK**-complete problem, but their problem is unnatural, and is defined in terms of the class itself. They also tried to modify the reductions from the statistical setting so that they apply to the perfect setting, but their idea works only in certain cases (e.g., when the underlying problem has a proof with perfect completeness). Bellare and Rogaway [2] showed other basic results about **NIPZK**, but their notion of zero knowledge allows simulation in expected (as opposed to strict) polynomial time. This notion is disadvantageous, especially when non-interactive protocols are executed as sub-protocols. The literature offers a variety of **NIPZK** proofs for specific problems (cf. [2,3,27]) and other results about **NIPZK** proofs that apply to problems with special properties (cf. [27–29]).

1.3. *Organization*

We use standard definitions, to be found in Sect. 2. In Sect. 3 we present the error shifting technique and use it to obtain a **NIPZK**-complete problem. In Sect. 4 we apply this technique to the interactive setting, where we obtain a hard problem. In Sect. 5 we show some applications of these results.

2. Preliminaries

We study complexity classes of *promise problems* [6], which are a generalization of languages. Formally, $\Pi \stackrel{\text{def}}{=} \langle \Pi_{\text{Y}}, \Pi_{\text{N}} \rangle$ is a *promise problem* if $\Pi_{\text{Y}} \cap \Pi_{\text{N}} = \emptyset$. The set Π_{Y} contains the YES instances of Π, and the set Π_{N} contains the NO instances of Π. We define $\overline{\Pi} \stackrel{\text{def}}{=} \langle \Pi_{\text{N}}, \Pi_{\text{Y}} \rangle$. Any language L can be defined as a promise problem $\langle L, \overline{L} \rangle$.

As in the study of statistical zero-knowledge, promise problems will be defined in terms of circuits. A circuit $X : \{0,1\}^m \to \{0,1\}^n$ is a boolean function, encoded in some way (see e.g. [24]), but we mainly treat X as a distribution, namely the output distribution of the circuit when the input to the circuit is uniformly distributed. Thus, given a set T, the probability $\Pr[X \in T]$ equals $\Pr_r[X(r) \in T]$, where r is uniformly chosen from $\{0,1\}^m$. The statistical distance between circuits, or more generally, the *statistical distance* between two discrete distributions X and Y, is defined as $\Delta(\text{X}, \text{Y}) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{\alpha} |\Pr[X = \alpha] - \Pr[Y = \alpha]|$.

2.1. *Protocols and Proofs*

We study both interactive and non-interactive perfect zero-knowledge proofs, using standard definitions [8]. We start with the definition of a non-interactive protocol, which we customize for the context of zero-knowledge proofs.

Definition 2.1 (Non-interactive protocols). A *non-interactive protocol* $\langle c, P, V \rangle$ is a triplet (or simply a pair $\langle P, V \rangle$, making c implicit), where P and V are functions, and $c \in \mathbb{N}$. We use r_P to denote the random input to P. The *interaction* between P and V on *common input* x is the following random experiment.

1. Uniformly choose r_P and a *common random string* $r_I \in \{0,1\}^{|x|^c}$.
2. Let $\pi = P(x, r_I; r_P)$, and let $m = V(x, r_I, \pi)$.
3. Output $\langle x, r_I, \pi, m\rangle$.

We call $\langle P, V\rangle(x) \stackrel{\text{def}}{=} \langle x, r_I, \pi, m\rangle$ the *view* of V on x. We say that V *accepts* x (respectively, *rejects* x) in $\langle P, V\rangle(x)$ if $m = \texttt{accept}$ (respectively, $m = \texttt{reject}$).

Definition 2.1 considers a deterministic verifier V. Non-interactive protocols where the verifier is probabilistic can be transformed into ones where the verifier is deterministic while preserving many of the properties of the original protocol [17]. The definition of interactive protocols is a simple extension of the above, except that there is no common random string, V has random input r_V, and P and V exchange messages until one of them accepts, rejects, or fails. Formally,

Definition 2.2 (Interactive protocols). An interactive protocol is a pair $\langle P, V\rangle$ of functions. The *interaction* between P and V on common input x is the following random experiment.

1. Let r_P and r_V be random inputs to P and V, respectively.
2. repeat the following for $i = 1, 2, \ldots$
   (a) If i is odd, let $m_i = P(x, m_1, \ldots, m_{i-1}; r_P)$.
   (b) If i is even, let $m_i = V(x, m_1, \ldots, m_{i-1}; r_V)$.
   (c) If $m_i \in \{\texttt{accept}, \texttt{reject}, \texttt{fail}\}$, then exit loop.

Each interaction yields a *transcript* $\langle x, m_1, \ldots, m_n; r_V\rangle$, and the strings m_i are called *messages*. The probability space containing all the transcripts is called *the view of* V *on* x, and is denoted $\langle P, V\rangle(x)$. We say that V *accepts* x if $m_i = \texttt{accept}$ for an even i.

We say that $\langle P, V\rangle$ is *constant round* if there is a constant c such that in any interaction the number of messages exchanged in at most c. We say that $\langle P, V\rangle$ is *public coin* for V if for any $n \in \mathbb{N}$ and any transcript ending with verifier message m_n, the randomness r_V accessed by V is $m_2 m_4 \cdots m_{n-2}$, and for all even $i \le n-2$ the length of m_i is a function of only $x, m_1, \ldots, m_{i-1}$. We say that $\langle P, V\rangle$ is *public coin* if $\langle P^*, V\rangle$ is public coin for any P^*.

A *proof* for a problem is a protocol that admits certain properties with respect to the problem. Informally, the verifier is efficient, with high probability it accepts YES instances of the problem, and with low probability it accepts NO instances (even if a computationally unbounded prover is cheating). In the following definition the difference between these probabilities is expressed via a non-negligible function c.

Definition 2.3 (Non-interactive proofs). A non-interactive protocol $\langle c, P, V\rangle$ is a *non-interactive proof* for a problem Π if there is a constant $a \in \mathbb{N}$ and $c(n), s(n) : \mathbb{N} \to [0,1]$ such that $1 - c(n) \ge s(n) + 1/n^a$ for every n, and the following conditions hold.

- Efficiency: V runs in time polynomial in $|x|$.
- Completeness: for all $x \in \Pi_{\text{YES}}$ we have $\Pr_{r_I, r_P}[V(x, r_I, P(x, r_I; r_P)) = \texttt{accept}] \ge 1 - c(|x|)$.

- Soundness: for any $x \in \Pi_{\mathrm{N}}$ and every function P^* we have $\Pr_{r_I, r_P}[V(x, r_I, P^*(x, r_I; r_P)) = \mathtt{accept}] \leq s(|x|)$.

The function c is called the *completeness error*, and the function s is called the *soundness error*. We say that $\langle P, V \rangle$ has *perfect completeness* if $c = 0$.

Although the completeness and soundness errors are defined using functions, in both the interactive and the non-interactive model our reductions will actually use $c = s = \frac{1}{3}$. This is without loss of generality because the reductions consider honest verifiers and therefore the errors can be reduced via parallel repetition.

Interactive proofs are defined from interactive protocols in exactly the same way, except that there is no reference string. Formally,

Definition 2.4 (Interactive proofs). Let $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ be a problem, and let $\langle P, V \rangle$ be an interactive protocol. We say that $\langle P, V \rangle$ is an *interactive proof* for Π if there is a, and $c(n), s(n) : \mathbb{N} \to [0, 1]$ such that $1 - c(n) > s(n) + 1/n^a$ for any n, and the following conditions hold.

- Efficiency: V is a probabilistic Turing machine whose running time over the entire interaction is polynomial in $|x|$ (this implies that the number of messages exchanged is polynomial in $|x|$).
- Completeness: if $x \in \Pi_{\mathrm{Y}}$, then V accepts in $\langle P, V \rangle(x)$ with probability at least $1 - c(|x|)$. The probability is over r_P and r_V (the randomness for P and V, respectively).
- Soundness: if $x \in \Pi_{\mathrm{N}}$, then for any function P^* V accepts in $\langle P^*, V \rangle(x)$ with probability at most $s(|x|)$. The probability is over the randomness r_P of P and r_V for V.

2.2. *Zero Knowledge*

We proceed to the definition of zero knowledge. Intuitively, a protocol is zero knowledge if the view of the verifier can be produced by the verifier itself, without help from the prover. This is formalized using the notion of a polynomial-time simulator that creates this view. Sequences of distributions $\{D(x)\}_{x \in T}$ and $\{D'(x)\}_{x \in T}$, called *ensembles*, are identically distributed if $D(x)$ and $D'(x)$ are identically distributed for all $x \in T$. We define zero knowledge with respect to simulators that do not fail, but in Sect. 5 we give a definition that allows failure, and show that the two are equivalent (for certain properties).

Definition 2.5 (Non-interactive zero-knowledge protocols). A non-interactive protocol $\langle P, V \rangle$ is *perfect zero knowledge* (**NIPZK**) for a problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ if there is a probabilistic Turing machine S running in strict polynomial time, called the *simulator*, such that the ensembles

$$\left\{\langle P, V \rangle(x)\right\}_{x \in \Pi_{\mathrm{Y}}} \quad \text{and} \quad \left\{S(x)\right\}_{x \in \Pi_{\mathrm{Y}}}$$

are identically distributed. If these ensembles are statistically indistinguishable, then $\langle P, V \rangle$ is a non-interactive *statistical zero-knowledge* (**NISZK**) protocol for Π. Sim-

ilarly, if the ensembles are computationally indistinguishable, then $\langle P, V\rangle$ is non-interactive *computational zero-knowledge* (**NICZK**) protocol for Π. The class of problems possessing **NIPZK** (respectively, **NISZK**, **NICZK**) proofs is also denoted **NIPZK** (respectively, **NISZK**, **NICZK**).

This definition can be extended to the interactive setting in the natural way. In the following, S^{V^*} denotes oracle access of S to the Turing machine V^*.

Definition 2.6 (Zero-knowledge protocols). A protocol $\langle P, V\rangle$ for a problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ is *perfect* (respectively, *statistical, computational*) *zero knowledge* if there is a probabilistic oracle Turing machine S running in strict polynomial time, called *the simulator*, such that for every probabilistic Turing machine V^* running in strict polynomial time we have

$$\left\{\left\langle P, V^*\right\rangle(x)\right\}_{x\in\Pi_{\mathrm{Y}}} \quad \text{and} \quad \left\{S^{V^*}(x)\right\}_{x\in\Pi_{\mathrm{Y}}}$$

are identically distributed (respectively, statistically indistinguishable, computationally indistinguishable.) The class of problems having perfect (respectively, statistical, computational) zero-knowledge protocols is denoted **PZK** (respectively, **SZK**, **CZK**.) When the above ensembles are indistinguishable for $V^* = V$ we say that $\langle P, V\rangle$ is *honest verifier, perfect* (respectively, *statistical, computational*) *zero knowledge*, and we denote the respective classes by **HVPZK**, **HVSZK**, and **HVCZK**.

3. A Complete Problem for NIPZK

In this section we introduce the *error-shifting technique* and use it to obtain the first complete problem for the class of problems admitting non-interactive perfect zero-knowledge (**NIPZK**) proofs. The proof system that we obtain has interesting characteristics, which we discuss later. We start with motivation, and give formal definitions and proofs in Sect. 3.1.

We describe STATISTICAL DISTANCE FROM UNIFORM (SDU), the **NISZK**-complete problem of [11], and explain why the reduction and the protocol for this problem cannot be applied to **NIPZK**. Instances of SDU are circuits that represent distributions, namely the output distribution of the circuit when the input to the circuit is uniformly distributed. Specifically, YES instances are circuits representing a distribution that is close to uniform, and NO instances are circuits representing a distribution that is far from uniform.

Definition 3.1. $\mathrm{SDU} \stackrel{\mathrm{def}}{=} \langle \mathrm{SDU_Y}, \mathrm{SDU_N}\rangle$, where

$$\mathrm{SDU_Y} = \left\{X \mid \Delta(\mathrm{X}, \mathrm{U_n}) < 1/n\right\},$$
$$\mathrm{SDU_N} = \left\{X \mid \Delta(\mathrm{X}, \mathrm{U_n}) > 1 - 1/n\right\},$$

X is a circuit with n output bits, and $\mathrm{U_n}$ is the uniform distribution on $\{0,1\}^n$.

The reduction of [11] (based on [30]) reduces any **NISZK** problem Π to SDU through a sequence of reductions. The part of this reduction that we modify is as follows. Let x be an instance of Π and let $\langle P, V \rangle$ be a **NISZK** proof for Π with a simulator S. The instance x is reduced to a circuit X which executes $S(x)$ and obtains a transcript. The transcript contains a simulated message of the prover and a simulated reference string. If the verifier accepts in this transcript, then X outputs the simulated reference string. Otherwise, X outputs the all-zero string. Intuitively, this reduction works because if x is a YES instance, then the simulated reference string is almost uniformly distributed, and thus X is a YES instance of SDU. Conversely, if x is a NO instance, then the verifier rejects on most reference strings, and thus X is a NO instance of SDU.

When we apply the reduction of [11] to **NIPZK** problems Π, and x is a YES instance, the output of S perfectly simulates the reference string. Thus, we expect to obtain a circuit X that represents the uniform distribution. However, if Π does not have perfect completeness, then the verifier may reject x, which skews the distribution represented by X. This will cause problems later, when we try to construct a proof system and a simulator for the complete problem. We overcome this issue using the *error shifting technique*.

The Error Shifting Technique In its most general form, the error shifting technique *shifts into the protocol errors that would otherwise become simulation errors*. This description is very loose, but we chose it because our technique can be applied in different contexts, and in each of these contexts it takes a different form. The following application will clarify our technique.

► *The first step of the error shifting technique* is to identify where the simulation error comes from. In our case, if the verifier rejects, then the circuit X does not represent the uniform distribution. Thus, the error comes from the completeness error of the proof of the underlying problem. Since we need to shift this error forward, we first separate it by adding an extra output bit to the circuit X. That is, X executes the simulator and outputs the simulated reference string followed by an extra bit. This bit takes the value 1 if the verifier accepts, and 0 if the verifier rejects.

► *The second step of the error shifting technique* is to shift the error forward, to the completeness or the soundness error of the protocol. In our case, from the circuit X to the protocol for our complete problem. This step is not trivial because we cannot just use the protocol of [11] for SDU. Specifically, in this protocol the prover sends a string r, and the verifier accepts if $X(r)$ equals the reference string. A simple analysis can show that even if we adapt this idea to our modified circuit, then we will get a simulation error. Thus, we modify this protocol by starting with the simulator, and constructing the prover based on the simulator. Informally, the simulator samples the circuit X, and the verifier accepts if the extra bit in this sample is 1. The prover simply mimics the simulator. This shifts the error from X to the completeness error of the new protocol. We make this intuition formal in the next section.

3.1. *A Complete Problem for **NIPZK***

In this section we formalize the intuition given in the previous section, thus proving that UNIFORM is **NIPZK**-complete. Our proof system has interesting characteristics, which we discuss after proving that UNIFORM is hard for **NIPZK**.

Theorem 3.2. UNIFORM (UN) *is* **NIPZK**-complete.

Recall that instances of UNIFORM are circuits X. Essentially, as a YES instance X represents the uniform distribution, and as a NO instance X has a small range. However, recall that X also has an extra rightmost output bit. To formally describe these properties, we use the convention that $n+1$ denotes the number of output bits of X. We use T_X to denote the outputs of X that end with the bit 1. Formally, $T_X \stackrel{\text{def}}{=} \{x1 | \exists r \text{ s.t. } X(r) = x1\}$, where $x1$ denotes the concatenation of the string x with the bit 1. Also, we use X' to denote the distribution on the n-bit prefix of the output of X. That is, X' is obtained by picking a random input r, computing $X(r)$, and taking the n-bit prefix of $X(r)$. As we shall see, when X is a YES instance of UNIFORM, the zero knowledge and completeness properties would imply that T_X is large and X' is the uniform distribution. Conversely, when X is a NO instances of UNIFORM, the soundness property would imply that $|T_X|$ is small.

The problem UNIFORM is defined in terms of T_X and X'. Formally, given a circuit X with $n+1$ output bits, we say that X is *β-negative* if $|T_X| \leq \beta \cdot 2^n$. That is, T_X has at most $\beta \cdot 2^n$ elements. We say that X is *α-positive* if X' is the uniform distribution on n bits and $\Pr_r[X(r) \in T_X] \geq \alpha$. This notion is not symmetric to that of β-negative, but it does imply that T_X has at least $\alpha \cdot 2^n$ elements.

Definition 3.3. The problem UNIFORM is defined as $\text{UN} \stackrel{\text{def}}{=} \langle \text{UN}_\text{Y}, \text{UN}_\text{N} \rangle$, where

$$\text{UN}_\text{Y} = \{X | X \text{ is } 2/3\text{-positive}\}, \quad \text{and}$$
$$\text{UN}_\text{N} = \{X | X \text{ is } 1/3\text{-negative}\}.$$

The constants $2/3$ and $1/3$ come from the completeness and soundness errors of the underlying proof, and as we mentioned in Sect. 2, these can be obtained from the definitions using repetition.

Proceeding to the completeness result, we recall that proving that a problem is complete for a given class requires proving that the problem is hard for the class (that is, any problem in the class reduces to this problem) and that it is in the class. Thus, we first show that the reduction from the previous section reduces every **NIPZK** problem to UNIFORM.

Lemma 3.4. UNIFORM *is* **NIPZK**-hard.

Proof. Let $\Pi = \langle \Pi_\text{Y}, \Pi_\text{N} \rangle$ be a **NIPZK** problem. Fix a non-interactive protocol $\langle P, V \rangle$ for Π with completeness and soundness errors $1/3$. Let r_I denote the common reference string in $\langle P, V \rangle$, and fix c such that $|r_I| = |x|^c$ for every $x \in \Pi_\text{Y} \cup \Pi_\text{N}$. Fix a simulator S for $\langle P, V \rangle$. Let $\ell \in \mathbb{N}$ such that the randomness of S on inputs of length n is of length at most n^ℓ. Let S' denote a circuit that on input $x \in \Pi_\text{Y} \cup \Pi_\text{N}$ and r_S of length $|x|^\ell$ outputs $S'(r_S) \stackrel{\text{def}}{=} S(x; r_S)$.

We show that Π Karp-reduces to UNIFORM. That is, we define a polynomial-time Turing machine that on input $x \in \Pi_\text{Y} \cup \Pi_\text{N}$ outputs a circuit $X : \{0,1\}^{|x|^\ell} \to \{0,1\}^{|x|^c+1}$ such that if $x \in \Pi_\text{Y}$, then $X \in \text{UN}_\text{Y}$, and if $x \in \Pi_\text{N}$, then $X \in \text{UN}_\text{N}$. On input

r_S of length $|x|^\ell$ the circuit X executes $S'(r_S)$ and obtains $S(x; r_S) = \langle x, r'_I, \pi \rangle$. If $V(x, r'_I, \pi) = \texttt{accept}$, then X outputs the string $r'_I 1$ (i.e., the concatenation of r'_I and 1), and otherwise it outputs $r'_I 0$.

Now we analyze our reduction. Let $x \in \Pi_{\mathrm{Y}}$, and let X be the output of the above reduction on x. We show that X is $2/3$-positive. Consider the distribution on the output $\langle x, r'_I, \pi \rangle$ of $S(x)$. Since $S(x)$ and $\langle P, V \rangle(x)$ are identically distributed, r'_I is uniformly distributed. Thus, X' (i.e., the distribution on the first $|x|^c$ output bits of X) is uniformly distributed. It remains to show that $\Pr[X \in T_X] \geq 2/3$. This immediately follows from the perfect zero knowledge and completeness properties of $\langle P, V \rangle$. That is, the output of S is identically distributed to $\langle P, V \rangle(x)$, and V accepts in $\langle P, V \rangle$ with probability at least $2/3$.

Let $x \in \Pi_{\mathrm{N}}$, and let X be the output of the above reduction on x. We show that X is $1/3$-negative. Assume towards contradiction that $|T_X| > 2^{|x|^c}/3$. We define a prover P^* that behaves as follows on CRS r_I. If $r_I 1 \in T_X$, then there is an input r_S to X such that $X(r_S) = r_I 1$. By the construction of X, there is randomness r_S for the simulator such that $S(x; r_S) = \langle x, r_I, \pi \rangle$, and $V(x, r_I, \pi) = 1$. In this case P^* sends π to V. If $r_I 1 \notin T_X$, then P^* fails. Notice that P^* makes V accept on any r_I such that $r_I 1 \in T_X$. Since $|T_X| > 2^{|x|^c}/3$, and since r_I is uniformly chosen in $\langle P^*, V \rangle$, the probability that $r_I 1 \in T_X$ is strictly greater than $1/3$. Thus, V accepts in $\langle P^*, V \rangle(x)$ with probability strictly greater than $1/3$, and contradiction to the soundness error of $\langle P, V \rangle$. Hence, X is $1/3$-negative. □

It remains to prove that UNIFORM is in **NIPZK**. We remark that our proof is unusual in the sense that we construct the prover and the verifier based on the simulator, and it is possible that on YES instances there are prover messages that will make the verifier accept, but instead the prover is sending a message that will make the verifier reject.

Lemma 3.5. UNIFORM *has a* **NIPZK** *proof with a deterministic verifier.*

Proof. Our prover and verifier for UNIFORM are based on the simulator, but describing the simulator before the proof is somewhat counter intuitive. Thus, we start with the proof. Given a circuit X with $n+1$ output bits, we use B_y to denote the set of all strings $\hat{r}$ for which the n-bit prefix of $X(\hat{r})$ is $y \in \{0,1\}^n$. On input $X : \{0,1\}^\ell \to \{0,1\}^{n+1}$ and common reference string $r_I \in \{0,1\}^n$ the prover P picks π uniformly from B_{r_I}. Such a π exists when $X \in \mathrm{UN}_{\mathrm{Y}}$ because X' (i.e., the distribution on the first n bits of X) is the uniform distribution. The deterministic verifier accepts if $X(\pi) = r_I 1$, and rejects otherwise.

Our prover is based on the following simulator. Let S be a probabilistic polynomial-time Turing machine that on input X uniformly picks $\pi' \in \{0,1\}^\ell$, and computes $z = X(\pi')$. The simulator assigns the n-bit prefix of z to r'_I (i.e., the simulated reference string), and outputs $\langle X, r'_I, \pi' \rangle$. Let $X \in \Pi_{\mathrm{Y}}$. We show that S perfectly simulates $\langle P, V \rangle$. Consider the distribution $S(X)$ on simulated transcripts $\langle X, r'_I, \pi' \rangle$, and the distribution $\langle P, V \rangle(X)$ on the view $\langle X, r_I, \pi \rangle$ of V. Since X' is uniformly distributed over $\{0,1\}^n$, the string r'_I obtained by the simulator is uniformly distributed over $\{0,1\}^n$. Since r_I is uniformly distributed, r'_I and r_I are identically distributed. It remains to show that π and π' are identically distributed conditioned on $r_I = r'_I$. For any simulated reference

string r'_I, the randomness π' chosen by the simulator is uniformly distributed in $B_{r'_I}$. Similarly, for any reference string r_I the message π of the prover is a string chosen uniformly from B_{r_I}. Hence, conditioned on $r_I = r'_I$, the strings π and π' are identically distributed. We conclude that $S(X)$ and $\langle P, V\rangle(X)$ are identically distributed for any $X \in \Pi_{\mathrm{Y}}$.

Turning our attention to the completeness property, we show that V accepts X with probability at least $2/3$. By the zero knowledge property, the output $\langle X, r'_I, \pi'\rangle$ of $S(X)$ is identically distributed to the view $\langle X, r_I, \pi\rangle$ of V on X. Thus, it is enough to show that when choosing a transcript $\langle X, r'_I, \pi'\rangle$ according to $S(x)$, the probability that $V(X, r'_I, \pi') = 1$ is at least $2/3$. Since S uniformly chooses π', and since X is $2/3$-positive, the probability that $X(r) \in T_X$ is at least $2/3$. Thus, the probability that the suffix of $X(r)$ is 1 is at least $2/3$. Hence, V accepts X with probability at least $2/3$. The soundness property follows easily. Let $X \in \mathrm{UN}_{\mathrm{N}}$. Since X is $1/3$-negative, $|T_X| \leq 1/3 \cdot 2^n$. Since r_I is uniformly distributed, the probability that $r_I 1 \in T_X$ is at most $1/3$. Hence, if $X \in \mathrm{UN}_{\mathrm{N}}$, then V accepts X with probability at most $1/3$. □

Theorem 3.2 follows from Lemmas 3.4 and 3.5.

4. A Hard Problem for Public-Coin PZK Proofs

This section shows a hard problem for the class of problems admitting public-coin, honest-verifier perfect zero-knowledge (**HVPZK**) proofs. This is achieved by removing the assumption on perfect completeness from the reduction of [26]. Our problem was used in [18] to study the round complexity of perfect zero-knowledge proofs and to prove an equivalence between zero knowledge and instance-dependent commitment schemes. Notice that since **PZK** $\subseteq$ **HVPZK**, our problem is also hard for public-coin **PZK** proofs.

For motivation, we start by describing STATISTICAL DISTANCE (SD), the complete problem of [26] for **SZK**. Instances of this problem are pairs $\langle X, Y\rangle$ of circuits. As YES instances, X and Y represent statistically close distributions, and as NO instances, X and Y are represent statistically far distributions. Specifically, $\mathrm{SD} \stackrel{\mathrm{def}}{=} \mathrm{SD}^{1/3,2/3}$, where $\mathrm{SD}^{\alpha,\beta}$ is defined as follows:

Definition 4.1. $\mathrm{SD}^{\alpha,\beta} \stackrel{\mathrm{def}}{=} \langle \mathrm{SD}^{\alpha,\beta}_{\mathrm{Y}}, \mathrm{SD}^{\alpha,\beta}_{\mathrm{N}}\rangle$, where

$$\mathrm{SD}^{\alpha,\beta}_{\mathrm{Y}} = \{\langle X_0, X_1\rangle |\ \Delta(\mathrm{X}_0, \mathrm{X}_1) \leq \alpha\}, \quad \text{and}$$
$$\mathrm{SD}^{\alpha,\beta}_{\mathrm{N}} = \{\langle X_0, X_1\rangle |\ \Delta(\mathrm{X}_0, \mathrm{X}_1) \geq \beta\}.$$

We remark that SD and $\overline{\mathrm{SD}}$ are referred to in the literature as the same problem because both of them are complete for **SZK** and reduce to each other. The reduction of [26] takes any problem that admits a public-coin, honest-verifier statistical zero-knowledge (**HVSZK**) proof and reduces it to SD. The issue with this reduction is that, when we apply it to the class of problems admitting public-coin, honest-verifier perfect zero-knowledge (**HVPZK**) proofs, we get a pair of circuits $\langle X_0, X_1\rangle$ that, as YES instances,

are only statistically close, but not identically distributed (unless the problem admits a proof with perfect completeness). This is unnatural because the closeness between X_0 and X_1 reflects the closeness of the simulation. Thus, in the perfect setting we expect X_0 and X_1 to be *identically distributed*, as in $\mathrm{SD}^{0,1/2}$.

Definition 4.2. $\mathrm{SD}^{0,1/2} \stackrel{\text{def}}{=} \langle \mathrm{SD}_{\mathrm{Y}}^{0,1/2}, \mathrm{SD}_{\mathrm{N}}^{0,1/2} \rangle$, where

$$\mathrm{SD}_{\mathrm{Y}}^{0,1/2} = \left\{ \langle X_0, X_1 \rangle \mid \Delta(\mathrm{X}_0, \mathrm{X}_1) = 0 \right\}, \quad \text{and}$$
$$\mathrm{SD}_{\mathrm{N}}^{0,1/2} = \left\{ \langle X_0, X_1 \rangle \mid \Delta(\mathrm{X}_0, \mathrm{X}_1) \geq 1/2 \right\}.$$

In the next section we describe the reduction to SD in more detail and show that, essentially, $\mathrm{SD}^{0,1/2}$ is hard for the class of problems admitting public-coin **HVPZK** proofs.

4.1. *A Hard Problem for Public-Coin **HVPZK** Proofs*

We show that, essentially, $\mathrm{SD}^{0,1/2}$ is hard for the class of problems admitting public-coin **HVPZK** proofs. This is done by applying the error shifting technique to the reduction of [26], which we now describe.

Let Π be a problem with a public-coin **HVPZK** proof $\langle P, V \rangle$ and a simulator S. Given a string x, we use $v \stackrel{\text{def}}{=} v(|x|)$ to denote the number of rounds in the interaction between P and V on input x. That is, in round i the prover P sends m_i and V replies with a random string r_i, until P sends its last message m_v, and V accepts or rejects. We denote the output of $S(x)$ by $\langle x, m_1, r_1, \ldots, m_v \rangle$. The reduction of [26] maps instances x of Π to pairs of circuits $\langle X', Y' \rangle$. These circuits are constructed from the circuits X_i and Y_i, defined as follows. The circuit X_i chooses randomness, executes $S(x)$ using this randomness, and outputs the simulated transcript, truncated at the ith round. That is, X_i obtains $\langle x, m_1, r_1, \ldots, m_v \rangle$, and outputs $\langle m_1, r_1, \ldots, m_i, r_i \rangle$. The circuit Y_i is defined exactly the same, except that it replaces r_i with a truly random string r_i'.

- $X_i(r)$: execute $S(x; r)$ to obtain $\langle x, m_1, r_1, \ldots, m_v \rangle$. Output $\langle m_1, r_1, \ldots, m_i, r_i \rangle$.
- $Y_i(r, r_i')$: execute $S(x; r)$ to obtain $\langle x, m_1, r_1, \ldots, m_v \rangle$. Output $\langle m_1, r_1, \ldots, m_i, r_i' \rangle$.

Notice that X_i and Y_i represent the same distribution when x is a YES instance. This is so because $S(x)$ perfectly simulates the view of the verifier, and therefore r_i is uniformly distributed, just like r_i'. Using $\otimes$ to denote the concatenation of circuits, let $X = X_1 \otimes \cdots \otimes X_v$. That is, X executes all the circuits X_i and outputs the concatenation of their outputs. Similarly, let $Y = Y_1 \otimes \cdots \otimes Y_v$. Again, X and Y are identically distributed when x is a YES instance. Now, the pair $\langle X', Y' \rangle$ is defined from $\langle X, Y \rangle$ as follows. The circuit Y' outputs the output of Y followed by 1. The circuit X' outputs the output of X followed by the output of Z, where Z is the circuit that outputs 1 if with high probability $S(x)$ outputs accepting transcripts, and 0 otherwise. Notice that Z can achieve this by running independent executions of $S(x)$ and estimating the probability that $S(x)$ output an accepting transcript.

The reduction of [26] does not apply to public-coin **HVPZK** proofs (unless we assume perfect completeness) because on YES instances x it is possible that V rejects x,

which would make the circuit Z output 0 with non-zero probability, and this leads to a non-zero statistical distance between X' and Y'. We overcome this issue using the error shifting technique.

Recall that the first step of the error shifting technique is to identify where the simulation error comes from. In this case, the error comes from the circuit Z. Since we need to shift this error forward, instead of including Z in the circuits X' and Y', we separate the error and map instances x of Π to triplets $\langle X, Y, Z\rangle$. Thus, if x is a YES instance, then X and Y are identically distributed, and Z outputs 1 with high probability. Such a triplet would be a YES instance of our hard problem. Similarly, by the simulator analysis from [26] (cf. [1,7,16,25]), if x is a NO instance, then either X and Y are statistically far, or Z outputs 0 with a high probability. Such a triplet would be a NO instance of our hard problem.

Lemma 4.3. *For any problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ possessing a public-coin* **HVPZK** *proof there is a Karp reduction mapping strings x to circuits $\langle X, Y, Z\rangle$ with the following properties.*

- *If $x \in \Pi_{\mathrm{Y}}$, then $\Delta(\mathrm{X}, \mathrm{Y}) = 0$ and $\Pr[Z = 1] \geq 2/3$.*
- *If $x \in \Pi_{\mathrm{N}}$, then $\Delta(\mathrm{X}, \mathrm{Y}) \geq 1/2$ or $\Pr[Z = 1] \leq 1/3$.*

Our hard problem can be defined as $\mathrm{SD}^{0,1/2}\wedge$ CAPP, where $\wedge$ denotes the AND of two promise problems, and CAPP is known as CIRCUIT APPROXIMATION PROBABILITY PROBLEM [24]. Recall that the AND of two promise problems Π and Γ is defined as $\Pi \wedge \Gamma \stackrel{\text{def}}{=} \langle(\Pi \wedge \Gamma)_{\mathrm{Y}}, (\Pi \wedge \Gamma)_{\mathrm{N}}\rangle$, where

$$(\Pi \wedge \Gamma)_{\mathrm{Y}} = \left\{\langle x, y\rangle | x \in \Pi_{\mathrm{Y}} \wedge y \in \Gamma_{\mathrm{Y}}\right\} \quad \text{and}$$
$$(\Pi \wedge \Gamma)_{\mathrm{N}} = \left\{\langle x, y\rangle | x \in \Pi_{\mathrm{N}} \vee y \in \Gamma_{\mathrm{N}}\right\}.$$

Instances of CAPP are circuits Z such that, $\Pr[Z = 1] \geq 2/3$ if Z is a YES instance, and $Pr[Z = 1] \leq 1/3$ if Z is a NO instance. CAPP is a complete promise problem for **BPP** (when considering **BPP** as a class of promise problems). We refer to $\mathrm{SD}^{0,1/2}\wedge$ CAPP as IDENTICAL DISTRIBUTIONS (ID).

The second step of the error shifting technique is to shift the error forward, to the completeness or soundness error of the protocol. However, we do not have a **HVPZK** proof for IDENTICAL DISTRIBUTIONS, and even $\mathrm{SD}^{0,1/2}$ is not known to have one (this was an open question in [26]). Thus, we show that given an arbitrary zero-knowledge protocol for $\mathrm{SD}^{0,1/2}$, the error can be shifted from the circuit Z to this protocol. In particular, this shows that any perfect zero-knowledge (**PZK**) proof for $\mathrm{SD}^{0,1/2}$ can be converted to a **PZK** proof for IDENTICAL DISTRIBUTIONS. Furthermore, we will preserve all the properties of the original protocol.

The error is shifted as follows. Let $\langle P, V\rangle$ be an arbitrary zero-knowledge protocol for $\mathrm{SD}^{0,1/2}$. We construct a new protocol $\langle P', V'\rangle$ on instances $\langle X, Y, Z\rangle$ of ID (instead of a pair $\langle X, Y\rangle$ of $\mathrm{SD}^{0,1/2}$). We let $P' = P$ and define V' just like V, except that before the protocol begins, V' estimates the value of $\Pr[Z = 1]$ and rejects if this value is at most 1/3. If V' did not reject, then P' and V' execute $\langle P, V\rangle$ on input $\langle X, Y\rangle$. Analyzing this protocol is straightforward. Notice that V' is very unlikely to reject if

$\Pr[Z = 1] \geq 2/3$, and that if the protocol continues, then either $\langle X, Y, Z\rangle$ is a YES instance of our hard problem and $\Delta(\mathrm{X}, \mathrm{Y}) = 0$, or $\langle X, Y, Z\rangle$ is a NO instance of our hard problem and $\Delta(\mathrm{X}, \mathrm{Y}) \geq 1/2$. Hence, in this case the behavior of P' and V' on instances of our hard problem is identical to the behavior of P and V on instances of $\mathrm{SD}^{0,\frac{1}{2}}$. The following theorem follows.

Theorem 4.4. *If* $\mathrm{SD}^{0,1/2}$ *has a public-coin* **HVPZK** *proof, then* ID *is complete for public-coin* **HVPZK**.

5. Applications

Our error shifting technique and hard problem were used in [18] to study perfect zero-knowledge proofs. In this section we show two additional applications of our results. The first one shows an equivalence between two notions of simulation. The second shows that, under certain conditions, non-interactive perfect zero-knowledge (**NIPZK**) proofs are closed under the OR operator.

5.1. *Obtaining Simulators that Do not Fail*

Zero-knowledge protocols have been defined in the literature with respect to simulators that are either allowed or not allowed to fail (also known as *abort*). We show that these notions are equivalent for honest-verifier zero knowledge. Our transformation shifts the simulation error into the completeness error and therefore does not preserve perfect completeness.

We first recall that the definitions of zero knowledge used in this paper (Definitions 2.5 and 2.6) require that the output of the simulator be "close" to the view of the verifier. A relaxation of this notion due to [5] allows the simulator to fail with probability at most $\frac{1}{2}$, and requires that, conditioned on non-failure, the output of the simulator be "close" to the view of the verifier. Notice that the constant $\frac{1}{2}$ is arbitrary as any non-negligible error probability can be reduced via repetition. The formal definition follows.

Definition 5.1 (Zero-knowledge protocols with simulators that can fail). A protocol $\langle P, V\rangle$ for a problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ is *perfect* (respectively, *statistical, computational*) *zero knowledge* if there is a probabilistic oracle Turing machine S running in polynomial time, called *the simulator*, such that for every probabilistic Turing machine V^* running in polynomial time the following holds:

1. For all $x \in \Pi_{\mathrm{Y}}$ we have $\Pr[S^{V^*}(x) = \texttt{fail}] \leq \frac{1}{2}$, where the probability is over the randomness of S and V^*.
2. Letting $\hat{S}^{V^*}(x)$ denote the distribution on the output of $S^{V^*}(x)$ conditioned on $S^{V^*}(x) \neq \texttt{fail}$, the following ensembles are identically distributed (respectively, statistically indistinguishable, computationally indistinguishable)

$$\left\{\langle P, V^*\rangle(x)\right\}_{x\in\Pi_{\mathrm{Y}}} \quad \text{and} \quad \left\{\hat{S}^{V^*}(x)\right\}_{x\in\Pi_{\mathrm{Y}}}.$$

It is well-known that in the statistical and the computational settings, a simulator S that is allowed to fail can be converted to a simulator S' that is not allowed to fail. On

common input x this can be done simply by running $|x|$ executions of $S(x)$, each with a fresh random input, and outputting the first non-fail output. If all executions fail, then $S(x)$ simply outputs `null`, but since this happens with probability at most $1/2^n$, the error that the `null` message introduces into the simulation is negligible. Thus, $S(x)$ is indistinguishable from the view of the verifier. Clearly, this simple idea does not apply to the perfect setting. In fact, since the simulation error is increased, this idea suggests that perhaps by allowing the simulator to fail, the prover may leak some knowledge to the verifier. By using the error shifting technique, we overcome this issue and show that the two notions of simulation are equivalent. We only consider the interactive setting, but the idea applies also to the non-interactive setting.

Lemma 5.2. *A problem Π has an honest-verifier perfect zero-knowledge proof according to Definition 2.6 if and only if it has an honest-verifier perfect zero-knowledge proof according to Definition 5.1.*

Proof. Trivially, Definition 2.6 implies Definition 5.1. In the forward direction, let Π be a problem with a perfect zero-knowledge protocol $\langle P, V\rangle$ and a simulator S that fails with probability at most $\frac{1}{2}$. The first step of the error shifting technique is to identify where the error is coming from and isolate it. In this case, the error comes from the failure probability of the simulator, and it is already separated from the output of the simulator. Hence, we proceed to the next step of the error shifting technique. That is, we shift the error into the protocol.

On input x we define a new prover P' whose first step is to run $|x|$ executions of $S(x)$. If $S(x)$ fails in all $|x|$ executions, then $P'(x)$ sends `null` to the verifier V and the protocol terminates. Otherwise, it behaves just like $P(x)$. The new simulator S' for $\langle P', V\rangle$ is modified to run $|x|$ executions of the original simulator $S(x)$. If all executions fail, then just like P', it sends `null` to the verifier V and the protocol terminates. Otherwise, one of the outputs of $S(x)$ is not `fail`, and S' outputs the first such non-fail output.

We analyze the new simulator S'. Consider all sufficiently long $x \in \Pi_Y$. The first observation is that $S'(x)$ never fails. The second observation is that both P' and S' send to V the message `null` with the same probability. Conditioned on S' not sending this message, the output of $S'(x)$ is identically distributed to the output of $S(x)$, which, by Definition 5.1, is identically distributed to the view $\langle P, V\rangle(x)$ of the honest verifier. Conditioned on P' not sending the `null` message, $\langle P', V\rangle(x)$ and $\langle P, V\rangle(x)$ are identically distributed because P' behaves just like P. Thus, $S(x)$ and $\langle P', V\rangle(x)$ are identically distributed. We conclude that the two notions are equivalent. □

5.2. *Under Certain Restrictions **NIPZK** is Closed Under the `OR` Operator*

In this section we prove a partial result towards showing that **NIPZK** is closed under the `OR` operator. We make strong conditions on the soundness and completeness error of a proof for the underlying problem. This illuminates the difficulties of working with perfect zero-knowledge proofs. No such closure result is known in the case of non-interactive statistical zero-knowledge (**NISZK**) proofs (cf. [11,30]).

Before we present our lemma, recall that a complexity class $\mathcal{C}$ is closed under the `OR` operator (denoted $\vee$) if for any two problems $\Pi, \Gamma \in \mathcal{C}$ we have $\Pi \vee \Gamma \in \mathcal{C}$, where

$\Pi \vee \Gamma \stackrel{\text{def}}{=} \langle (\Pi \vee \Gamma)_{\mathrm{Y}}, (\Pi \vee \Gamma)_{\mathrm{N}} \rangle$,

$$(\Pi \vee \Gamma)_{\mathrm{Y}} = \left\{ \langle x, y \rangle | x \in \Pi_{\mathrm{Y}} \vee y \in \Gamma_{\mathrm{Y}} \right\}, \quad \text{and}$$
$$(\Pi \vee \Gamma)_{\mathrm{N}} = \left\{ \langle x, y \rangle | x \in \Pi_{\mathrm{N}} \wedge y \in \Gamma_{\mathrm{N}} \right\}.$$

Notice that, since we are working with promise problems, in the definition of $(\Pi \vee \Gamma)_{\mathrm{Y}}$, when one of x, y is not a YES instance, the intention is that this element is a NO instance of either Π or Γ. Our lemma follows.

Lemma 5.3. *Let Π and Γ be* **NIPZK** *problems. Consider the reduction from these problems to instances of* UNIFORM, *and denote by X circuits with $n+1$ output bits obtained by the reduction. If as* YES *instances the circuits X are* 1*-positive, and as* NO *instances the circuits X are $2^{-(1+n/2)}$-negative, then $\Pi \vee \Gamma \in$* **NIPZK**.

Proof. We show that $\Pi \vee \Gamma$ Karp-reduces to UNIFORM.

Let X and Y be the circuits obtained by reducing instances of Π and Γ, respectively, to UNIFORM as assumed in the hypothesis of the lemma. We denote by X' and Y' the n-bit prefix of X and Y, respectively. Our Karp reduction builds a circuit Z from X and Y. The circuit Z outputs $n+1$ bits using the following computation:

1. If the suffix bit of both X and Y is 1, then Z outputs $X' \oplus Y'$, followed by 1.
2. If the suffix bit of exactly one of X and Y is 1, then Z outputs the output of that circuit.
3. If the suffix bit of both X and Y is 0, then Z outputs the all-zero string.

To complete the proof, we need to show that $Z \in \mathrm{UN}_{\mathrm{Y}}$ if at least one of X and Y is a YES instance, and that $Z \in \mathrm{UN}_{\mathrm{N}}$ if both X and Y are NO instances. Since YES instances are 1-positive, if at least one of X and Y is a YES instance, then the suffix bit of this instance is 1 and the n-bit prefix of this instance is uniformly distributed over $\{0,1\}^n$. Without loss of generality this instance is X. Thus, depending on the output of Y, the circuit Z either outputs the output of X, or it outputs $X' \oplus Y'$, followed by 1. In both cases the suffix bit of Z is 1 and Z' is uniformly distributed over $\{0,1\}^n$, where Z' is the n-bit prefix of Z. Hence, Z is 1-positive.

We turn our attention to NO instances of Π and Γ. As usual, T_X denotes the set of outputs of X whose rightmost bit is 1. We define T_Y and T_Z analogously for Y and Z, respectively. Since NO instances are $2^{-(1+n/2)}$-negative, $|T_X| \leq 2^n \cdot 2^{-(1+n/2)}$. Similarly, $|T_Y| \leq 2^n \cdot 2^{-(1+n/2)}$. By the construction of Z it follows that $|T_Z| \leq |T_X| \cdot |T_Y| + |T_X| + |T_Y| \leq 2^n/4 + 2^{-n/2} \leq 1/3 \cdot 2^n$. This implies that Z is $1/3$-negative. The lemma follows. □

References

[1] W. Aiello, J. Håstad, Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.* **42**(3), 327–345 (1991)

[2] M. Bellare, P. Rogaway, Noninteractive perfect zero-knowledge. Unpublished manuscript, June 1990

[3] M. Blum, A. De Santis, S. Micali, G. Persiano, Noninteractive zero-knowledge. *SIAM J. Comput.* **20**(6), 1084–1118 (1991)

[4] G. Brassard, C. Crépeau, M. Yung, Everything in NP can be argued in perfect zero-knowledge in a bounded number of rounds (extended abstract), in *EUROCRYPT '89: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology* (Springer, New York, 1990), pp. 192–195

[5] I. Damgård, O. Goldreich Avi Wigderson, Hashing functions can simplify zero-knowledge protocol design (too). Technical report RS-94-39, BRICS, November 1994

[6] S. Even, A.L. Selman, Y. Yacobi, The complexity of promise problems with applications to public-key cryptography. *Inf. Control* **61**(2), 159–173 (1984)

[7] L. Fortnow, The complexity of perfect zero-knowledge, in *Advances in Computing Research*, vol. 5, ed. by S. Micali (JAC Press, 1989), pp. 327–343

[8] O. Goldreich, *Foundations of Cryptography*, vol. 1 (Cambridge University Press, Cambridge, 2001)

[9] O. Goldreich, S. Micali, A. Wigderson, Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM* **38**(3), 691–729 (1991)

[10] O. Goldreich, A. Sahai, S.P. Vadhan, Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge, in *STOC '98: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (ACM, New York, 1998), pp. 399–408

[11] O. Goldreich, A. Sahai, S.P. Vadhan, Can statistical zero knowledge be made non-interactive? Or on the relationship of SZK and NISZK, in *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology* (Springer, London, 1999), pp. 467–484

[12] O. Goldreich, S.P. Vadhan, Comparing entropies in statistical zero-knowledge with applications to the structure of SZK, in *IEEE Conference on Computational Complexity* (1999), pp. 54–73

[13] S. Goldwasser, M. Sipser, Private-coins versus public-coins in interactive proof systems, in *Advances in Computing Research*, vol. 5, ed. by S. Micali (JAC Press, 1989), pp. 73–90

[14] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems. *SIAM J. Comput.* **18**(1), 186–208 (1989)

[15] J. Groth, R. Ostrovsky, A. Sahai, Perfect non-interactive zero knowledge for NP, in *Proceedings of Eurocrypt 2006*. LNCS, vol. 4004 (Springer, Berlin, 2006), pp. 339–358

[16] I. Haitner, O. Reingold, S.P. Vadhan, H. Wee, Inaccessible entropy, in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, ed. by M. Mitzenmacher, Bethesda, MD, USA, May 31—June 2, 2009 (ACM, New York, 2009), pp. 611–620

[17] L. Babai, S. Moran, Arthur-merlin games: a randomized proof system and a hierarchy of complexity classes. *J. Comput. Syst. Sci.* **36**, 254–276 (1988)

[18] L. Malka, Instance-dependent commitment schemes and the round complexity of perfect zero-knowledge proofs. *Electron. Colloq. Comput. Complex.* **15**, 068 (2008)

[19] S. Micali, R. Pass, Local zero knowledge, in *STOC '06: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, 2006), pp. 306–315

[20] M. Naor, R. Ostrovsky, R. Venkatesan, M. Yung, Perfect zero-knowledge arguments for p using any one-way permutation. *J. Cryptol.* **11**(2), 87–108 (1998)

[21] M.-H. Nguyen, S. Vadhan, Zero knowledge with efficient provers, in *STOC '06: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2006), pp. 287–295

[22] T. Okamoto, On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.* **60**(1), 47–108 (2000)

[23] S.J. Ong, S.P. Vadhan, An equivalence between zero knowledge and commitments, in *TCC*, ed. by R. Canetti. Lecture Notes in Computer Science, vol. 4948 (Springer, Berlin, 2008)

[24] C.H. Papadimitriou, *Computational Complexity*, vol. 10 (Addison Wesley, Reading, 1993)

[25] E. Petrank, G. Tardos, On the knowledge complexity of NP, in *FOCS* (1996), pp. 494–503

[26] A. Sahai, S.P. Vadhan, A complete problem for statistical zero-knowledge. *J. ACM* **50**(2), 196–249 (2003)

[27] A. De Santis, G. Di Crescenzo, G. Persiano, The knowledge complexity of quadratic residuosity languages. *Theor. Comput. Sci.* **132**(1–2), 291–317 (1994)

[28] A. De Santis, G. Di Crescenzo, G. Persiano, Randomness-efficient non-interactive zero-knowledge (extended abstract), in *Automata, Languages and Programming*, (1997), pp. 716–726

[29] A. De Santis, G. Di Crescenzo, G. Persiano, On NC^1 boolean circuit composition of non-interactive perfect zero-knowledge, in *MFCS* (2004), pp. 356–367

[30] A. De Santis, G. Di Crescenzo, G. Persiano, M. Yung, Image density is complete for non-interactive-SZK (extended abstract), in *Automata, Languages and Programming, 25th International Colloquium, ICALP'98*, ed. by K.G. Larsen, S. Skyum, G. Winskel, Aalborg, Denmark, July 13–17, 1998. Lecture Notes in Computer Science, vol. 1443 (Springer, Berlin, 1998), pp. 784–795

[31] M. Tompa, H. Woll, Random self-reducibility and zero-knowledge interactive proofs of possession of information, in *FOCS '87: 28th Annual Symposium on Foundations of Computer Science*, Los Angeles, California, USA, 12–14 October 1987 (IEEE Press, New York, 1987), pp. 472–482

[32] S.P. Vadhan, A study of statistical zero-knowledge proofs. PhD thesis, MIT (1999)

[33] J. Watrous, Zero-knowledge against quantum attacks, in *STOC '06: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, 2006), pp. 296–305