Unconditional Characterizations of Non-Interactive Zero-Knowledge

Rafael Pass[1*] and abhi shelat[2**]

[1] MIT CSAIL, pass@csail.mit.edu
[2] IBM Zurich Research, abhi@zurich.ibm.com

Abstract. Non-interactive zero-knowledge (NIZK) proofs have been investigated in two models: the *Public Parameter* model and the *Secret Parameter model*. In the former, a public string is "ideally" chosen according to some efficiently samplable distribution and made available to both the Prover and Verifier. In the latter, the parties instead obtain correlated (possibly different) private strings. To add further choice, the definition of zero-knowledge in these settings can either be *non-adaptive* or *adaptive*.
In this paper, we obtain several *unconditional* characterizations of computational, statistical and perfect NIZK for all combinations of these settings. Specifically, we show:

In the secret parameter model, $\mathbf{NIZK} = \mathbf{NISZK} = \mathbf{NIPZK} = \mathbf{AM}$.
In the public parameter model,
- ▷ for the non-adaptive definition, $\mathbf{NISZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$,
- ▷ for the adaptive one, it also holds that $\mathbf{NISZK} \subset \mathbf{BPP}/1$,
- ▷ for computational NIZK for "hard" languages, one-way functions are both *necessary* and *sufficient*.

From our last result, we arrive at the following *unconditional* characterization of computational NIZK in the public parameter model (which complements well-known results for interactive zero-knowledge):

> Either NIZK proofs exist only for "easy" languages (i.e., languages that are not hard-on-average), or they exist for all of $\mathbf{AM}$ (i.e., all languages which admit non-interactive proofs).

1 Introduction

A zero-knowledge proof system is a protocol between two parties, a Prover, and a Verifier, which guarantees two properties: a malicious Prover cannot convince the Verifier of a false theorem; a malicious Verifier cannot learn anything from an interaction beyond the validity of the theorem.

Non-interactive zero-knowledge (NIZK) was proposed by Blum, Feldman, and Micali [BFM88] to investigate the *minimal* interaction necessary for zero-knowledge proofs. To achieve the absolute minimal amount of interaction —that

---

* Research supported by an Akamai Presidential Fellowship.
** This research was completed while at MIT CSAIL.

is, a single message from the Prover to the Verifier— some *setup* assumptions are provably necessary [GO94]. These setup assumptions can be divided into two groups:

1. **Public Parameter Setup.** The originally proposed setup is the *Common Random String Model* in which a uniformly random string is made available to both the Prover and Verifier. Many NIZK schemes have been implemented in this model [SMP87, BFM88, FLS90, DMP88, BDMP91, KP98, DCO$^+$01]. A slight relaxation of this model is the *Public Parameter model*, also known as the *Common Reference String Model*, in which a string is "ideally" chosen according to some polynomial-time samplable distribution and made available to both the Prover and Verifier. Such a setup can be used to select —say— safe primes, group parameters, or public keys for encryption schemes, etc. See for example [Dam00, CLOS02].
2. **Secret Parameter Setup.** Cramer and Damgård [CD04] explicitly introduce the Secret Parameter setup model in which the Prover and Verifier obtain correlated (possibly different) private information.
   More generally, the secret parameter model encompasses the *Pre-processing Model* in which the Prover and Verifier engage in an arbitrary interactive protocol, at the end of which, both Prover and Verifier receive a private output. (This follows because any arbitrary protocol for pre-processing can be viewed as a polynomial-time sampler from a well-defined distribution.) Such a setup model is studied in [KMO89, DMP88, Dam93].

The above setup models can be *implemented* in a variety of ways, which may or may not require their own independent assumptions (For example, secure two-party computations protocols can be used to pick a random string.) In this paper we defer the discussion of *how* trusted setups are implemented, and choose instead to focus on the relative *power* of the models.

We restrict our study to the simplest setting in which only a single theorem is proven. Also, we consider security against unbounded provers. (That is, we consider *proof systems* as opposed to *argument systems*.) Following similar studies in the interactive setting —see for example [Vad99, SV03, Vad04]— we allow the honest prover algorithm to be inefficient (although some of our constructions have efficient prover algorithm for languages in **NP**).

Our investigation also considers both *adaptive* and *non-adaptive* definitions of zero-knowledge for non-interactive proofs. Briefly, the difference between these two is that the adaptive variant guarantees that the zero-knowledge property holds even if the theorem statement is chosen *after* the trusted setup has finished, whereas the non-adaptive variant does not provide this guarantee.

1.1 Our Results

Secret Parameter Model One suspects that the secret-parameter setup is more powerful than its public-parameter counterpart. Indeed, in game theory, a well-known result due to Aumann [Aum74] states that players having access to *correlated* secret strings can achieve a larger class of equilibria, and in particular, better payoffs, than if they only share a common public string. As we shall see, this intuition carries over in a strong way to the cryptographic setting. But first, we show that,

> **Informal Theorem** [Upper bound] In the secret parameter model, non-interactive *perfect* zero-knowledge proofs exist *unconditionally* for all languages in **AM**.

This result is obtained by combining the work of [FLS90] with an adaptation of Kilian's work on implementing commitments using oblivious transfer [Kil88].

Previously, for general **NP** languages, only *computational* NIZK proof systems were known in the secret-parameter setup model [DMP88, FLS90, KMO89, DFN05]. Furthermore, these systems relied on various computational assumptions, such as the existence of one-way permutations. Recently, Cramer and Damgård [CD04] constructed statistical NIZK proofs in this model for *specific* languages related to discrete logarithms. (On the other hand, their results apply to an unbounded number of proofs, whereas ours do not.)

As a corollary of our result, we obtain a complete characterization of computational, statistical and perfect NIZK in the secret parameter model. Namely, we show that **NIP** = **NIZK** = **NISZK** = **NIPZK** = **AM**, where **NIP** denotes the class of languages having non-interactive proofs, and **NIZK**, **NISZK** and **NIPZK** denotes the classes of languages having non-interactive computational, statistical and perfect zero-knowledge proofs.

PUBLIC PARAMETER MODEL: STATISTICAL NIZK We next turn our attention to the public parameter model, and show that, in contrast to the Secret Parameter model, statistical NIZK proofs for **NP**-complete languages are unlikely to exist.[3]

> **Informal Theorem** [Lower bound] In the public parameter model, non-interactive statistical (non-adaptive) zero-knowledge proof systems only exist for languages in **AM** $\cap$ **coAM**.

Previously, Aiello and Håstad [AH91] showed a similar type of lower bound for *interactive* zero-knowledge proofs. Although their results extend to the case of NIZK in the *common random string* model, they do not extend to the *general* public parameter model.[4] Indeed, our proof relies on different (and considerably simpler) techniques.
In the case of statistical *adaptive* NIZK, we present a stronger result.

[3] This follows because unless the polynomial hierarchy collapses, **NP** is not contained in **AM** $\cap$ **coAM** [BHZ87].

[4] This follows because the definition of zero-knowledge requires the simulator to output the random coins of the Verifier, and this is essential to the result in [AH91]. In contrast, the definition of NIZK in the Public Parameter model does not require the Simulator to output the random coins used by the trusted-party to generate the public parameter.

> **Informal Theorem** [Lower bound] Non-interactive statistical adaptive zero-knowledge proof systems only exist for languages in $\mathbf{BPP}/1$ (i.e., the class of languages decidable in probabilistic polynomial time with *one* bit of advice, which depends only on the length of the instance).

By an argument of Adleman, this in particular means that all languages which have statistical adaptive NIZK in the public-parameter model can be decided by polynomial-sized circuits.

We note that a similar strengthening for the non-adaptive case is unlikely, as statistical non-interactive zero-knowledge proof systems for languages which are conjectured to be "hard" are known (e.g., see [GMR98]).

PUBLIC PARAMETER MODEL: COMPUTATIONAL NIZK Due to the severe lower bounds for statistical NIZK, we continue our investigation by considering computational NIZK in the public parameter model. We first show that one-way functions are both necessary and sufficient in the public parameter model.

> **Informal Theorem** [Upper bound] If (non-uniform) one-way functions exist, then computational NIZK proof systems in the public parameter model exist for every language in **AM**.

> **Informal Theorem** [Lower bound] The existence of computational NIZK systems in the public parameter model for a hard-on-average language implies the existence of (non-uniform) one-way functions.

Our upper bound, which applies to the stronger adaptive definition, improves on the construction of Feige, Lapidot, and Shamir [FLS90] which uses one-way permutations (albeit in the common random string model, whereas our construction requires a public parameter). Our lower bound, which applies to the weaker non-adaptive definition, was only known for interactive zero-knowledge proofs [OW93]. We therefore present a (quite) different and relatively simple direct proof for the case of NIZK in the public parameter model.

As a final point, by combining our last two theorems, we obtain the following unconditional characterization of computational NIZK proofs in the public parameter model:

> *Either NIZK proofs exist only for "easy" languages (i.e., languages that are not hard-on-average), or NIZK proofs exist unconditionally for every language in* **AM** *(i.e., for every language which admits a non-interactive proof).*

This type of "all-or-nothing" property was known for interactive zero-knowledge proofs, but not for NIZK since prior constructions of NIZK relied on one-way permutations.

ADDITIONAL CONTRIBUTIONS As already mentioned, some proofs in this paper extend previously known results for interactive zero-knowledge proofs to the non-interactive setting. We emphasize that our proofs are not mere adaptations of prior results — indeed the results of Aiello and Håstad and of Ostrovsky and

Wigderson are complicated and technically challenging. In contrast, in the non-interactive setting, we obtain equivalent results in a much simpler way. This suggests the use of non-interactive zero-knowledge as a "test-bed" for understanding the (seemingly) more complicated setting of interactive zero-knowledge.

1.2 Other Related Work

In terms of understanding NIZK, two prior works, [DCPY98] and [GSV99], offer complete problems for non-interactive statistical zero-knowledge. Both of these works apply to the non-adaptive definition and only the common *random* string model. We emphasize that these results do not directly extend to the more general public parameter model. In particular, complete problems for **NISZK** in the public parameter model are not known (see the remarks following Thm. 4).

As mentioned earlier, many prior works, e.g. [AH91, Oka96, SV03, GV98, Vad99], address the problem of obtaining unconditional characterizations of statistical zero-knowledge in the interactive setting. More recently, Vadhan [Vad04] also obtains unconditional characterizations of computational zero-knowledge.

OPEN QUESTIONS While our NIZK proof system in the secret parameter model has an efficient prover strategy, our proof system in the public parameter model does not. Indeed, resolving whether one-way functions suffice for *efficient-prover* NIZK systems is a long-standing open question with many important implications. A positive answer to this question would, for example, lead to the construction of CCA2-secure encryption schemes from any semantically-secure encryption scheme.

2 Definitions

We use standard notation for probabilistic experiments introduced in [GMR85], and abbreviate probabilistic polynomial time as p.p.t.

2.1 Non-interactive Proofs in the Trusted Setup model

In the trusted setup model, every non-interactive proof system has an associated distribution $\mathcal{D}$ over binary strings of the form (s_V, s_P). During a setup phase, a trusted party samples from $\mathcal{D}$ and privately hands the Prover s_P and the Verifier s_V. The Prover and Verifier then use their respective values during the proof phase. We emphasize that our definition only models *single-theorem* proof systems (i.e., after setup, only one theorem of a fixed size can be proven).[5]

Definition 1 (Non-Interactive Proofs in the Secret/Public Parameter Model). *A triple of algorithms, $(\mathcal{D}, P, V)$, is called a non-interactive proof system in the secret parameter model for a language L if the algorithm $\mathcal{D}$ is probabilistic polynomial-time, the algorithm V is a deterministic polynomial-time and there exists a negligible function μ such that the following two conditions hold:*

[5] While our definition only considers single-theorem proof systems, all of our results extend also to proof systems for an *a priori* bounded number of fixed-size statements.

– COMPLETENESS: *For every $x \in L$*

$$\Pr\left[(s_V, s_P) \leftarrow \mathcal{D}(1^x);\ \pi \leftarrow P(x, s_P)\ :\ V(x, s_V, \pi) = 1\right] \geq 1 - \mu(|x|)$$

– SOUNDNESS: *For every $x \notin L$, every algorithm B*

$$\Pr\left[(s_V, s_P) \leftarrow \mathcal{D}(1^x);\ \pi' \leftarrow B(x, s_P)\ :\ V(x, s_V, \pi') = 1\right] \leq \mu(|x|)$$

If $\mathcal{D}$ is such that s_V is always equal to s_P then we say that $(\mathcal{D}, P, V)$ is in the public parameter model.

Remark 1. In our definition, as with the original one in [BFM88], the Verifier is modeled by a deterministic polynomial time machine. By a standard argument due to Babai and Moran [BM88], this choice is without loss of generality since a probabilistic Verifier can be made to run deterministically through repetition and the embedding of the Verifier's random coins in the setup information.

Let **NIP** denote the class of languages having non-interactive proof systems. For the rest of this paper, we distinguish the secret parameter model from the public parameter model using the superscripts $^{\mathrm{SEC}}$ and $^{\mathrm{PUB}}$ respectively. We start by observing that $\mathbf{NIP}^{\mathrm{PUB}}$ and $\mathbf{NIP}^{\mathrm{SEC}}$ are equivalent. The proof appears in the full version.

Lemma 1. $\mathbf{AM} = \mathbf{NIP}^{\mathrm{PUB}} = \mathbf{NIP}^{\mathrm{SEC}}$

2.2 Zero Knowledge

We next introduce non-interactive zero-knowledge proofs. In the original *non-adaptive* definition of zero-knowledge from [BFM88], there is one simulator, which, after seeing the statement to be proven, generates both the public string and the proof at the same time. In a later *adaptive* definition from [FLS90], there are two simulators— the first of which must output a string before seeing any theorems. The stronger *adaptive* definition guarantees zero-knowledge *even* when the statements are chosen after the trusted setup has finished.[6] Here, we choose to present a weaker (and simpler) *adaptive* definition similar to the one used in [CD04]. The main reasons for this choice are that (a) a weaker definition only strengthens our lower bounds and (b) our definition is meaningful also for languages outside of **NP**, whereas the definitions of [FLS90, Gol04] only apply to languages in **NP**. Nevertheless, we mention that for languages in **NP**, our upper bounds (and of course the lower bounds) also hold for the stricter adaptive definitions of [FLS90, Gol04].

Definition 2 (Non-Interactive Zero-Knowledge in the Secret/Public Parameter Model). *Let $(\mathcal{D}, P, V)$ be an non-interactive proof system in the secret (public) parameter model for the language L. We say that $(\mathcal{D}, P, V)$ is non-adaptively zero-knowledge in the secret (public) parameter model if there exists*

[6] One might also study an adaptive notion of soundness for non-interactive proofs. We do not pursue this line since every sound non-interactive proof system can be made adaptively sound via parallel repetition.

a p.p.t. simulator S such that the following two ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of $|x|$)

$$\{(s_V, s_P) \leftarrow \mathcal{D}(1^n);\ \pi \leftarrow P(s_P, x)\ :\ (s_V, \pi)\ \}_{x \in L}$$
$$\{((s'_V, \pi') \leftarrow S(x)\ :(s'_V, \pi')\ \}_{x \in L}$$

We say that $(\mathcal{D}, P, V)$ is adaptively zero-knowledge *in the secret (public) parameter model if there exists two p.p.t. simulators S_1, S_2 such that the following two ensembles are computationally indistinguishable by polynomial-sized circuits.*

$$\{(s_V, s_P) \leftarrow \mathcal{D}(1^n);\ \pi \leftarrow P(s_P, x)\ :\ (s_V, \pi)\ \}_{x \in L}$$
$$\{(s'_V, \mathtt{aux}) \leftarrow S_1(1^n);\ \pi' \leftarrow S_2(x, \mathtt{aux})\ :(s'_V, \pi')\ \}_{x \in L}$$

We furthermore say that $(\mathcal{D}, P, V)$ is perfect (statistical) zero-knowledge *if the above ensembles are identically distributed (statistically close).*

For notation purposes, we will use **NIZK**, **NISZK**, and **NIPZK** to denote the class of languages having computational, statistical, and perfect non-interactive zero-knowledge proof systems respectively.

3 The Hidden Bits Model

In order to prove our main theorems, we first review the "hidden bits" model described in [FLS90]. In this model, the Prover and Verifier share a hidden string R, which only the Prover can access. Additionally, the Prover can selectively reveal to the Verifier any portion of the string R. Informally, a proof in the hidden bits model consists of a triplet (π, R_I, I) where I is a sequence of indicies, $I \subseteq \{1, 2, ..., |R|\}$ representing the portion of R that the prover wishes to reveal to the verifier, R_I is the substring of R indexed by I, and π is a proof string. For a formal definition of this model, see Goldreich [Gol01] from which we borrow notation.

The following theorem is shown by Feige, Lapidot and Shamir.

Theorem 1 ([FLS90]). *There exists a non-interactive perfect zero-knowledge proof system in the hidden bits model for any language in* **NP**.

We extend their result to any language in **AM** by using the standard technique of transforming an **AM** proof into the **NP** statement that "there exists a short Prover message which convinces the polynomial-time Verifier."

Theorem 2. *There exists a non-interactive perfect zero-knowledge proof system in the hidden bits model for any language in* **AM**.

Looking ahead, in Sect. 4 we extend Thm. 2 to show that the class of non-interactive perfect zero-knowledge proofs in the hidden bits model is in fact *equivalent* to **AM**.

4 The Secret Parameter Model

Feige, Lapidot and Shamir show how to implement the hidden-bits model with a one-way permutation in the public parameter model. Their implementation, however, degrades the quality of zero-knowledge — in particular, the resulting protocol is only computational zero-knowledge. Below, we show how to avoid this degradation in the secret parameter model.

Lemma 2. *Let (P, V) be a non-interactive perfect zero-knowledge proof system for the language L in the hidden bits model. Then, there exists a non-interactive perfect adaptive zero-knowledge proof system (P', V') for the language L in the secret parameter model. Furthermore if, (P, V) has an efficient prover, then (P', V') has one as well.*

Proof Sketch. We implement the hidden bits model by providing the Prover and Verifier correlated information about each bit of the hidden string. In particular, each bit is split into shares using a simple secret sharing scheme. The Prover is given *all* of the shares, while the Verifier is only given a random subset of them (which is unknown to the Prover). This is done in such a way that the Verifier has no information about the bit, but nonetheless, the Prover cannot reveal the bit in two different ways except with exponentially small probability. We note that this technique is reminiscent to the one used in [Kil88] to obtain commitments from oblivious transfer and to the one in [KMO89] to obtain NIZK with pre-processing (we remark that their resulting NIZK still requires additional computational assumptions, even when ignoring the assumptions necessary for their pre-processing). Our protocol is described in Fig. 1 and a complete proof is given in the full version. □

Armed with this Lemma, we can now prove our main theorem concerning non-interactive zero-knowledge in the secret parameter model.

Theorem 3. $\mathbf{NIP}^{\mathrm{SEC}} = \mathbf{NIZK}^{\mathrm{SEC}} = \mathbf{NISZK}^{\mathrm{SEC}} = \mathbf{NIPZK}^{\mathrm{SEC}} = \mathbf{AM}$

Proof. $\mathbf{NIPZK}^{\mathrm{SEC}} \subseteq \mathbf{NISZK}^{\mathrm{SEC}} \subseteq \mathbf{NIZK}^{\mathrm{SEC}} \subseteq \mathbf{NIP}^{\mathrm{SEC}}$ follows by definition. Lemma 1 shows that $\mathbf{NIP}^{\mathrm{SEC}} = \mathbf{AM}$, therefore, it suffices to show that $\mathbf{AM} \subseteq \mathbf{NIPZK}^{\mathrm{SEC}}$. This follows by combining Lemma 2 and Thm. 2. □

Related Characterizations We note that Lemma 2 also gives an upper bound on the class of perfect zero-knowledge proofs in the hidden bits model. As a corollary, we obtain the following characterization.

Corollary 1. *The class of perfect zero-knowledge proofs in the hidden bits model equals* **AM**.

5 The Public Parameter Model - Statistical NIZK

In this section we present severe lower bounds for the class of statistical NIZK in the public parameter model. (This stands in stark contrast to the secret

Proof System $(\mathcal{D}, P', V')$ – **NIZK in the Secret Parameter model**

Common Input: an instance x of a language L with witness relation R_L and 1^n: security parameter.

Private-output set-up: $\mathcal{D}(1^n) \rightarrow (s_P, s_V)$ proceeds as follows on input 1^n:

1. **(Pick a random string)** Sample m random bits, $\sigma = \sigma_1, \ldots, \sigma_m$.
2. **(Generate XOR shares)** For $i \in [1, m]$ and $j \in [1, n]$, sample a random bit τ_i^j. Let $\overline{\tau}_i^j = \sigma_i \oplus \tau_i^j$. (Notice that the n pairs $(\tau_i^j, \overline{\tau}_i^j)$ for $j \in [1, n]$ are n random "XOR shares" of the bit σ_i.)
3. **(Select half of each share)** For $i \in [1, m]$ and $j \in [1, n]$, sample a random bit b_n^j. Let ρ_i^j as follows:

$$\rho_i^j = \begin{cases} \tau_i^j, & \text{if } b_i^j = 0 \\ \overline{\tau}_i^j & \text{otherwise} \end{cases}$$

   (In other words, the values $\{\rho_i^j\}$ are randomly selected "halves" from each of the n XOR shares for σ_i.)
4. The private output s_P is the set of nm pairs $(\tau_i^j, \overline{\tau}_i^j)$ for $i, j \in [1, m] \times [1, n]$. Note that the string σ is easily derived from s_P.
5. The private output s_V is the set of nm pairs $\{(\rho_i^j, b_i^j)\}$ for $i, j \in [1, m] \times [1, n]$.

Prover algorithm: On input (x, s_P),

1. Compute $R = \sigma_1, \ldots, \sigma_m$ by setting $\sigma_i = \tau_i^1 \oplus \overline{\tau}_i^1$.
2. Run the algorithm $(\pi, R_I, I) \leftarrow P(x, R)$. Recall that the set R_I consists of bits $\{r_i \mid i \in I\}$ and I consists of indices in $[1, m]$.
3. Output $(\pi, R_I, I, \{o_i \mid i \in I\})$ where o_i denotes the opening of bit σ_i. That is, for all $i \in I$, o_i consists of all n shares $((\tau_i^1, \overline{\tau}_i^1), \ldots, (\tau_i^n, \overline{\tau}_i^n))$ of σ_i.

Verifier algorithm: On input $(x, s_V, \pi, R_I, I, \{o_i | i \in I\})$,

1. Verify that each opening in R_I is consistent with o_i and with s_V. That is, for $i \in I$, inspect the n pairs, $(\tau_i^1, \overline{\tau}_i^1), \ldots, (\tau_i^n, \overline{\tau}_i^n)$ in o_i, and check that for all $j \in [1, n]$, ρ_i^j is equal to either τ_i^j or $\overline{\tau}_i^j$ (depending on whether $b_i^j = 0$ or 1 respectively). If any single check fails, then reject the proof. Finally, check that $r_i = \tau_i^1 \oplus \overline{\tau}_n^1$.
2. Verify the proof by running $V(x, R_I, I, \pi)$ and accept if and only if V accepts.

Fig. 1. NIZK in the Secret Parameter model

parameter model, where statistical NIZK can be obtained for all of **AM**.) We first present a lower bound for statistical NIZK under the non-adaptive definition of zero-knowledge. We thereafter sharpen the bound under the more restrictive adaptive definition.

5.1 The Non-Adaptive Case

In analogy with the result by [AH91] for interactive zero-knowledge, we show that only languages in the intersection of **AM** and **coAM** have statistical NIZK proof systems in the public parameter model.

Theorem 4. *If L has a statistical non-adaptive NIZK proof system in the public parameter model, then $L \subseteq \mathbf{AM} \cap \mathbf{coAM}$.*

Proof Sketch. Let $(\mathcal{D}, P, V)$ be a statistical NIZK proof system in the public parameter for the language L with simulator S. We show that $L \in \mathbf{AM}$ and that $L \in \mathbf{coAM}$. The former statement follows directly from Lemma 1. To prove the latter one, we present a two-round proof system for proving $x \notin L$. (Note that by the results of [GS86, BM88] it is sufficient to present a two-round *private* coin proof system.)

Verifier Challenge:
1. Run the simulator $(\sigma_0, \pi') \leftarrow S(x)$ and the sampling algorithm $\sigma_1 \leftarrow D(1^{|x|})$ to generate public parameter strings σ_0 and σ_1.
2. Run V on input (σ_0, π') to check if the honest verifier accepts the simulated proof. If V rejects, then output "accept" and halt.
3. Otherwise, flip a coin $b \in 0, 1$ and send $\alpha = \sigma_b$ to the prover.

The Prover response:
1. Upon receiving an input string α, check if there exists a proof π which the honest verifier V accepts (i.e., $V(x, \alpha, \pi) = 1$).
2. If so, output $\beta = 0$; otherwise, output $\beta = 1$.

The Verifier acceptance condition:
1. Upon receiving string β, output "accept" if $\beta = b$, and reject otherwise.

COMPLETENESS We show that if $x \notin L$, then the Prover (almost) always convinces the Verifier. If the Verifier sent the string σ_0, the Prover always responds with $\beta = 0$, which makes the Verifier always accept. This follows since the Verifier only sends σ_0 if the simulated proof was accepting, which implies that there is at least one accepting proof of $x \in L$ for (P, V). If the Verifier sent the string σ_1, then by the soundness of (P, V), the probability (over the coins of the Verifier) that there exists a proof for $x \in L$ is negligible. Therefore, except with negligible probability, the Prover responds with $\beta = 1$ and the Verifier accepts.

SOUNDNESS Intuitively, this protocol relies on the same logic as the graph non-isomorphism protocol. If $x \in L$, then the (exponential time) Prover cannot distinguish whether α was generated by the simulator or by the sampler $\mathcal{D}$, and therefore can only convince the Verifier with probability 1/2. This follows from the statistical zero-knowledge property of (P, V). It only remains to show that the probability (over the random coins of the Verifier) that the Verifier accepts statements $x \in L$ in step (2), without further interaction, is negligible. This follows from the zero-knowledge (and completeness) property of (P, V). Otherwise, V would distinguish between simulated proofs and real ones (since by completeness, the honest prover P succeeds with high probability.) □

Remark 2. Using techniques from the proof of Thm. 4, one can show that the class $\mathbf{NISZK}^{\mathrm{PUB}}$ reduces to the problem of Statistical Difference, which is complete for **SZK** [SV03][7]. Thus, an alternative way to prove this theorem would be to present such a reduction and then invoke the results of [AH91].

[7] This should be contrasted with Statistical Difference from Random and Image Density, which are the complete problems for **NISZK** in the Common Random String model. These problems are not known to be reducible to Statistical Difference

5.2 The Adaptive Case

In this section we sharpen our results from the previous section when instead considering the adaptive variant of zero-knowledge.

Theorem 5. *If L has a non-interactive adaptive statistical zero-knowledge proof in the public parameter model, then $L \subset \mathbf{BPP}/1$.*

Proof Sketch. Let $(\mathcal{D}, P, V)$ be a non-interactive adaptive statistical zero-knowledge proof system for L with simulators S_1 and S_2.

We first observe that by the statistical zero-knowledge property, for every n for which L contains an instance of length n, the output of $S_1(1^n)$ must be statistically close to the output of $\mathcal{D}(1^n)$. This follows because the output of $S_1(1^n)$ is independent of the theorem statement.

This observation suggests the following probabilistic polynomial time decision procedure $D(x)$ for L, which obtains a one-bit non-uniform advice indicating whether L contains any instances of length $|x|$.

On input an instance x,
1. If the non-uniform advice indicates that L contains no instances of length $|x|$, directly reject.
2. Otherwise, run $(\sigma', \mathtt{aux}) \leftarrow S_1(1^{|x|})$ to generate a public parameter.
3. Run $\pi' \leftarrow S_2(x, \mathtt{aux})$ to produce a putative proof.
4. Run $V(x, \sigma', \pi')$ and accept iff V accepts.

Note that when $x \in L$, then D accepts with overwhelming probability due to the completeness and zero-knowledge property of $(\mathcal{D}, P, V)$. If $x \notin L$ and there are no instances of length $|x|$ in L, then D always rejects due to the non-uniform advice. It remains to show that when $x \notin L$, and there exists instances of length $|x|$ in L, then D rejects with high probability.

Assume, for sake of reaching contradiction, that there exists a polynomial $p(\cdot)$ such that for infinitely many lengths n, L contains instances of length n yet there exists an instance $x \notin L$ of length n, such that

$$\Pr\left[(\sigma', \mathtt{aux}) \leftarrow S_1(1^{|x|});\ \pi' \leftarrow S_2(x, \mathtt{aux})\ :\ V(x, \sigma', \pi') = 1\right] \geq \frac{1}{p(n)} \qquad (1)$$

We show how this contradicts the fact that the output of S_1 and $\mathcal{D}$ are statistically close (when L contains instances of length n). By the soundness of $(\mathcal{D}, P, V)$, there exists a negligible function μ such that for any unbounded prover P^*,

$$\Pr\left[\sigma \leftarrow \mathcal{D}(1^{|x|});\ \pi' \leftarrow P^*(x, \sigma)\ :\ V(x, \sigma, \pi') = 1\right] \leq \mu(|x|) \qquad (2)$$

Consider an exponential time non-uniform distinguisher C, which on input σ'' (and advice x), enumerates all proof strings π' to determine if any of them convince V to accept x. If so, C outputs 0, and otherwise outputs 1.

If σ'' is generated by S_1, then by (1), such a proof string π' exists with noticeable probability. On the other hand, if σ'' comes from $\mathcal{D}$, then by (2), such a proof string only exists with negligible probability. We conclude that C distinguishes the output of S_1 from that of $\mathcal{D}$ with a non-negligible advantage. $\square$

6 The Public Parameter Model - Computational NIZK

In this section we show that one-way functions are sufficient and necessary for computational NIZK for languages that are hard-on-average. Combining these two results, we obtain the following unconditional characterization : *Either* $\mathbf{NIZK}^{\mathrm{PUB}}$ *only contains "easy" languages (i.e., languages that are not hard-on-average), or it "hits the roof", (i.e., contains all of* **AM***).*

PRELIMINARIES Let us first define one-way functions and hard-on-average languages. As is standard in the context of zero-knowledge proofs, we define hardness in terms of infeasability for *non-uniform* p.p.t.

Definition 3 (One-way function). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called* one-way *if the following two conditions hold:*

- *Easy to compute: There exists a (deterministic) polynomial-time algorithm E such that on input x, E outputs $f(x)$.*
- *Hard to invert: For every non-uniform p.p.t. algorithm A, every sufficiently large integer n, and every polynomial $p(\cdot)$,*

$$\Pr\left[x \leftarrow \{0,1\}^n; y \leftarrow A(f(x)) \;:\; f(y) = f(x)\right] < \frac{1}{p(n)}$$

Definition 4 (Hard-on-average language). *A language L is hard-on-average if there exists a p.p.t. sampling algorithm G such that for every non-uniform p.p.t. algorithm A, every polynomial $p(\cdot)$, and every sufficiently large n,*

$$\Pr\left[x \leftarrow G(1^n) : \; A(x) \text{ correctly decides whether } x \in L\right] < \frac{1}{2} + \frac{1}{p(n)}$$

6.1 OWFs are Sufficient

We show how to implement the hidden bits model in the public-parameter model based on a one-way function. Recall that [FLS90] implements the hidden bits model using a one-way permutation and a hard-core predicate. The reason for using a one-way permutation is to give the Prover a short certificate for opening each bit *in only one way* (the certificate being the pre-image of the one-way permutation). A similar technique fails with one-way functions since a string may have either zero or many pre-images, and therefore a malicious Prover may be able to open some hidden bits as either zero or one.

Another approach would be to use a one-way function in order to construct a pseudo-random generator [HILL99], and then to represent a 0 value as a pseudo-random string and a 1 as a truly random string (in some sense, this technique is reminiscent of the one used by Naor for bit commitment schemes from pseudo-random generators [Nao91]). The Prover can thus open a 0 value by revealing a seed to the pseudo-random string. However, there is no way for the Prover to convince a Verifier that a string is truly random.

We overcome this problem by forming a reference string consisting of *pairs* of $2k$-bit strings, (α, β) in which exactly one of the two strings is pseudo-random while the other is truly random. More precisely, the 0-value is encoded as a pair in which α is generated pseudo-randomly by expanding a k bit seed into a $2k$ bit string, while β is chosen uniformly at random from $\{0,1\}^{2k}$. The 1-value is encoded the opposite way: α is chosen randomly, while β is generated pseudo-randomly. The Prover can now reveal a 0 or a 1 by revealing the seed for either α or β.

Lemma 3. *Assume the existence of one-way functions. Let (P, V) be a non-interactive (adaptive) zero-knowledge proof system for the language $L \in \mathbf{NP}$ in the hidden bits model. If P is an efficient prover, then, there exists a non-interactive (adaptive) zero-knowledge proof system (P', V') for the language L in the public parameter model.*

Proof Sketch. Let (P, V) be an NIZK proof system in the hidden bits model, let $G : \{0,1\}^k \to \{0,1\}^{2k}$ be a pseudo-random generator and let $L \in \mathbf{NP}$ be a language with witness relation R_L. Consider protocol (P', V') described in Fig. 2.

Completeness Completeness follows from the corresponding completeness of (P, V) and the fact that P' aborts only with negligible probability.

Soundness Assume for the moment that a cheating prover P'^* is only able to open R in one manner. In this case, the soundness of (P, V) carries over to (P', V') in the same way as in Lemma 2. All that remains is to show that R can only be opened in one way. Below, we argue that this happens with high probability.

Note that there are a maximum of 2^n pseudo-random strings in G's support. On the other hand, there are 2^{2n} strings of length $2n$. Therefore, a randomly sampled length-$2n$ string will be pseudo-random with probability at most 2^{-n}. Thus, for any pair (a_i, b_i), the probability that both values are pseudo-random is at most 2^{-n}. By the union bound, the probability that there is one such pair in s is upper-bounded by $n2^{-n}$.

Zero-knowledge We present a simulator $S' = S'_1, S'_2$ for $(\mathcal{D}, P', V')$ which uses the simulator S for (P, V) as a subroutine. First, $(s, \mathtt{aux}) \leftarrow S'_1(1^n)$ generates s as a sequence of pairs (α'_i, β'_i) in which *both* α'_i and β'_i are pseudo-random. The $\mathtt{aux}$ value contains all of the seeds, u_i, w_i, for the pseudo-random values α'_i and β'_i respectively. The simulator S'_2 works by running simulator $S(x)$ to generate $(\pi', R'_I, I) \leftarrow S(x)$ and then outputting $(\pi', R'_I, I, \{v'_i \mid i \in I\})$ where v'_i equals u_i if $r_i = 0$ and w_i otherwise. In order to show the validity of the simulation, consider the following four hybrid distributions.

- Let H_1 denote the ensemble (s, π) in which the honest Prover runs on a string s generated according to $\mathcal{D}$.
- Let H_2 denote the output of the above experiment with the exception that $\mathcal{D}$ provides all pre-images $\{v_i\}$ to an efficient prover algorithm P_{eff}, which also

Proof System $(\mathcal{D}, P', V')$ **– NIZK in the Public Parameter model**

Common Input: an instance $x \in L$ and a security parameter 1^n
Public Parameter set-up: $\mathcal{D}(1^n) \to s$, where $\mathcal{D}$ proceeds as follows :

1. Select m random bits $\sigma = \sigma_1, \ldots, \sigma_m$.
2. For each $i \in [1, m]$, generate two strings α_i and β_i as follows:
   $\alpha_i \leftarrow G(v_i)$ where v_i is a uniformly chosen string of length k.
   $\beta_i \leftarrow_r \{0,1\}^{2k}$
3. Let $\tau_i = \begin{cases} (\alpha_i, \beta_i) & \text{if } \sigma_i = 1 \\ (\beta_i, \alpha_i) & \text{otherwise} \end{cases}$
4. Output $s = \tau_1, \ldots, \tau_m$.

Prover's algorithm: On input x, s,

1. Compute $R = \sigma_1, \ldots, \sigma_m$ from s by the following procedure. Parse s into m pairs $(a_1, b_1), \ldots, (a_m, b_m)$. For each pair (a_i, b_i), determine (in exponential time) which of either a_i or b_i are pseudo-random (i.e, in the range of G). In the former case, set $\sigma_i = 0$, and in the latter, $\sigma_i = 1$, and let v_i denote the seed used to generate the pseudo-random value. If both a_i and b_i are in the range of G, then output `abort`.
2. Compute the lexographically first witness w satisfying $R_L(x, w)$.
3. Run the Prover algorithm $(\pi, R_I, I) \leftarrow P(x, w, R)$. Recall that the set R_I consists of bits $\{r_i \mid i \in I\}$ and I consists of indices in $[1, m]$.
4. Output $(\pi, R_I, I, \{v_i \mid i \in I\})$.

Verifier's algorithm: On input $(x, \pi, R_I, I, \{v_i \mid i \in I\})$

1. Verify each opening in R_I is consistent with s and v_i. Parse s into m pairs $(a_1, b_1), \ldots, (a_m, b_m)$. For each $i \in I$, run $t \leftarrow G(v_i)$ and if $t = a_i$, set $\sigma_i = 1$, if $t = b_i$, then set $\sigma_i = 0$ (if neither or both conditions are met, then reject the proof). Finally, verify that $r_i = \sigma_i$.
2. Run the Verifier algorithm $V(x, \pi, R_I, I)$ and accept iff V accepts.

Fig. 2. NIZK in the Public Parameter model

receives the lexographically first witness w for x and then only runs Step 3 and 4 of P''s algorithm.

– Let H_3 denote the output of the second experiment with the exception that s is generated by $S_1'(1^n)$, and that furthermore, $S_1'(1^n)$ gives either u_i or w_i (randomly chosen) to P_{eff} for all $i \in [1, m]$.
– Let H_4 denote the output of the third experiment with the exception that π is generated by $S_2'(x, \texttt{aux})$ and u_i, w_i in $\texttt{aux}$ is given to P_{eff}. Notice that this distribution corresponds exactly to the output of S'.

In the full version we show that the above hybrid distributions are all indistinguishable, which concludes the proof. □

Remark 3. Note that we explicitly require two properties from the NIZK proof system (P, V) in the hidden bits model: first, that P is an efficient Prover, and secondly, that the zero-knowledge property is defined for non-uniform distinguishers. Both of these requirements stem from the fact that the Prover in our new protocol is unbounded, which creates complications in the hybrid arguments.

Theorem 6. *If (non-uniform) one-way functions exist, then for both adaptive and non-adaptive definitions of zero-knowledge,* $\mathbf{NIZK}^{\mathrm{PUB}} = \mathbf{NIP}^{\mathrm{PUB}} = \mathbf{AM}$.

Proof. By Thm. 1 and Lemma 3, $\mathbf{NP} \subseteq \mathbf{NIZK}^{\mathrm{PUB}}$. Using techniques from the proof of Thm. 2, we can extend this result to show that $\mathbf{AM} \subseteq \mathbf{NIZK}^{\mathrm{PUB}}$. By definition, $\mathbf{NIZK}^{\mathrm{PUB}} \subseteq \mathbf{NIP}^{\mathrm{PUB}}$. Finally, by Lemma 1, $\mathbf{NIP}^{\mathrm{PUB}} = \mathbf{AM}$. □

6.2 OWFs are Necessary

We proceed to show that (non-uniform) one-way functions are *necessary* for non-interactive zero-knowledge for "hard" languages. This stands in contrast to the secret parameter model where unconditional results are possible.

Theorem 7. *If there exists a non-adaptive NIZK proof system for a hard-on-average language L, then (non-uniform) one-way functions exist.*

Proof Sketch. Let $(\mathcal{D}, P, V)$ be a non-adaptive NIZK system for L in the public parameter model and let S be the simulator for (P, V). Furthermore, suppose that L is hard-on-average for the polynomial-time samplable distribution G. Now, consider the following two distributions:

$$\{(s_V, s_P) \leftarrow \mathcal{D}(1^n), x \leftarrow G(1^n) \,:\, x, s_V\} \tag{3}$$

$$\{(s'_V, \pi) \leftarrow S(x, 1^n), x \leftarrow G(1^n) \,:\, x, s'_V\} \tag{4}$$

We show that the above distributions are (non-uniformly) computationally indistinguishable, but statistically "far". By a result of Goldreich [Gol90] (relying on [HILL99]) the existence of such distributions implies the existence of (non-uniform) one-way functions.

Claim. The distributions (3) and (4) are computationally indistinguishable.

Proof Sketch. We start by noting that conditioned on x being a member of language L, the above distributions are computationally indistinguishable by the zero-knowledge property of (P, V). It then follows from the hardness of L that the above distributions must be computationally indistinguishable, even without this restriction. □

Claim. The distributions (3) and (4) are *not* statistically indistinguishable.

Proof Sketch. We show that the distributions (3) and (4) are statistically "far" conditioned on instances $x \notin L$. It then follows from the fact that L is roughly balanced over G (due the hard-on-average property of L over G) that (3) and (4) are statistically "far" apart.

Note that on instances $x \notin L$, the soundness property of (P, V) guarantees that very few strings generated by $\mathcal{D}$ have proofs which are accepted by the

Verifier (otherwise, a cheating prover can, in exponential time, find such proofs and thereby violate the soundness condition). On the other hand, since L is hard-on-average, and since S runs in polynomial time, most of the strings s_V generated by S have proofs which are accepted by V (otherwise, S can be used to decide L). Therefore, the distributions (3) and (4) are statistically far apart, conditioned on instances $x \notin L$. $\square$ $\square$

ACKNOWLEDGMENTS We would like to thank Silvio Micali and the anonymous referees for their helpful suggestions.

References

[AH91] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci*, 42:327–345, 1991.

[Aum74] R. Aumann. Subjectivity and correlation in randomized strategies. *J. Math. Econ.*, 1:67–96, 1974.

[BM88] L. Babai and S. Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci*, 36(2):254–276, 1988.

[BDMP91] M. Blum, A. De Santis, S. Micali, and G. Persiano. Noninteractive zero-knowledge. *SIAM J. Computing*, 20(6):1084–1118, 1991.

[BFM88] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC 88*, pages 103–112, 1988.

[BHZ87] R. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[CLOS02] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC 02*, pages 494–503, 2002.

[CD04] R. Cramer and I. Damgård. Secret-key zero-knowledge and non-interactive verifiable exponentiation. In *TCC 04*, 2004.

[Dam93] I. Damgård. Non-interactive circuit based proofs and non-interactive perfect zero knowledge with preprocessing. In *EUROCRYPT 92*, pages 341–355, 1993.

[Dam00] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT 2000*, pages 418–430, 2000.

[DFN05] I. Damgård, N. Fazio, and A. Nicolosi. Secret-key zero-knowledge protocols for NP and applications to threshold cryptography. Manuscript, 2005.

[DCO+01] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and Amit Sahai. Robust non-interactive zero knowledge. *CRYPTO 01*, pages 566–598, 2001.

[DCPY98] A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung. Image density is complete for non-interactive-SZK. In *ICALP 98*, pages 784–795, 1998.

[DMP88] A. De Santis, S. Micali, and G. Persiano. Non-interactive zero-knowledge with preprocessing. In *CRYPTO 88*, pages 269–282, 1988.

[FLS90] U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero knowledge proofs based on a single random string. In *FOCS 90*, pages 308–317, 1990.

[GMR98] R. Gennaro, D. Micciancio, and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In *CCS 98*, pages 67–72, 1998.

[Gol90] O. Goldreich. A note on computational indistinguishability. *Inf. Process. Lett.*, 34(6):277–281, 1990.

[Gol01] O. Goldreich. *Foundations of Cryptography: Vol I.* 2001.

[Gol04] O. Goldreich. *Foundations of Cryptography: Vol II.* 2004.

[GO94] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *J. Crypt.*, 7(1):1–32, 1994.

[GSV99] 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.

[GV98] O. Goldreich and S. Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In *Computational Complexity*, 1998.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 85*, pages 291–304, 1985.

[GS86] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *STOC 86*, pages 59–68, 1986.

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[Kil88] J. Kilian. Founding cryptography on oblivious transfer. In *STOC 88*, pages 20–31, 1988.

[KMO89] J. Kilian, S. Micali, and R. Ostrovsky. Minimum resource zero-knowledge proofs. In *FOCS 89*, pages 474–479, 1989.

[KP98] J. Kilian and E. Petrank. An efficient non-interactive zero-knowledge proof system for NP with general assumptions. *J. Crypt.*, 11(1):1–27, 1998.

[Nao91] M. Naor. Bit commitment using pseudorandomness. *J. Crypt.*, 4(2):151–158, 1991.

[Oka96] T. Okamoto. On relationships between statistical zero-knowledge proofs. In *STOC 96*, pages 649–658, 1996.

[OW93] R. Ostrovsky and A. Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

[SV03] A. Sahai and S. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[SMP87] A. De Santis, S. Micali, and G. Persiano. Non-interactive zero-knowledge proof systems. In *CRYPTO 87*, pages 52–72, 1987.

[Vad99] S. Vadhan. *A Study of Statistical Zero-Knowledge Proofs.* PhD thesis, MIT, 1999.

[Vad04] S. Vadhan. An unconditional study of computational zero knowledge. In *FOCS 04*, pages 176–185, 2004.