Statistical Zero-Knowledge Arguments for NP Using Approximable-Preimage-Size One-Way Functions

Iftach Haitner* Ronen Shaltiel†

December 1, 2004

Abstract

A statistical zero knowledge argument for NP is a cryptographic primitive that allows a polynomial-time prover to convince another polynomial-time verifier of the validity of an NP statement. It is guaranteed that even an infinitely powerful verifier does not learn any additional information but the validity of the claim.

Naor et al. [NOVY98] showed how to implement such a protocol using any *one-way permutation*. We achieve such a protocol using any *approximable-preimage-size* one-way function. These are one-way functions with the additional feature that there is a feasible way to approximate the number of preimages of a given output. A special case is regular one-way functions where each output has the same number of preimages.

Our result is achieved by showing that a variant of the computationally-binding bit-commitment protocol of Naor et al. can be implemented using a any one-way functions with "sufficiently dense" output distribution. We construct such functions from approximable-preimage-size one-way functions using "hashing techniques" inspired by Hastad et al. [HILL98].

1 Introduction

1.1 Zero-Knowledge

A Zero-Knowledge proof of some statement (a notion introduced in [GMR89]) is a way for one player ("the prover") to convince another player ("the verifier") in the validity of a statement without revealing any additional information. We are interested in the setting where the proof is done interactively and the statement to be proven is the validity of an NP statement. The notion of zero-knowledge has become fundamental in Cryptography, and zero-knowledge protocols are used as a building block for many applications (c.f. [Gol02]). It is important to distinguish between two variants:

Zero-Knowledge Proofs: In this setup (introduced by Goldwasser et al. [GMR89]) the soundness of the proof is *information theoretic* meaning that even an infinitely powerful prover cannot convince the verifier of a false statement. However, the zero-knowledge guarantee is *computational*, meaning that a computationally-bounded verifier learns no additional information.

*Department of Computer Science, Weizmann Institute of Science. Rehovot, Israel.

†Department of Computer Science, University of Haifa, Israel. Part of this research was done while in the Weizmann Institute and supported by the Koshland Scholarship.

Zero-Knowledge Arguments: In this setup (introduced by Chaum et al. [BCC88]) the soundness of the proof is *computational* meaning that a computationally-bounded prover cannot convince the verifier of a false statement. However, the zero-knowledge guarantee is *information theoretic* meaning that even an infinitely powerful verifier cannot extract additional information. Here there is an additional distinction between *perfect* protocols (in which the verifier gets no information whatsoever) and *statistical* protocols (in which the verifier might get some information with negligible probability). [1]

A fundamental problem is to construct zero-knowledge protocols using as weak as possible assumptions. The existing protocols for both variants use a protocol by Goldreich et al. [GMW91]. This protocol reduces the task of zero-knowledge to that of "bit-commitment".

1.2 Bit-Commitment

A Bit-Commitment protocol is a protocol in which one party, "the sender", is giving the other party, "the receiver", a "sealed envelope" containing some secret bit. The protocol should be "hiding" meaning that the receiver learns nothing about this bit. It should also be "binding" meaning that in a later stage the sender can "open the envelope" and the receiver can be sure that its content did not change. Once again, there are two variants:

Computationally-hiding and Perfectly-binding: Here the hiding guarantee is *computational* meaning that a computationally-bounded receiver gains knowledge on the content of the envelope with at most negligible probability. The binding guarantee is information theoretic meaning that even an infinitely powerful sender cannot "change" the content of the envelope after it has been sealed.

Perfectly-hiding and Computationally-binding: Here the hiding guarantee is information theoretic meaning that even an infinitely powerful receiver learns nothing on the content of the envelope. The hiding guarantee is computational meaning that a computationally-bounded sender cannot "change" the content of the envelope. Again, there is an additional distinction. A protocol is *Statistically-hiding* if an infinitely powerful receiver learns unallowed information with at most negligible probability.

Interestingly, invoking the [GMW91] proof systems with the "right type" of bit-commitment yields the "right type" of zero-knowledge protocol. More precisely, invoking the protocol with a computationally-hiding perfectly-binding bit-commitment yields zero-knowledge proofs for NP. Invoking the protocol with a perfectly-hiding computationally-binding bit-commitment yields perfect zero-knowledge arguments, and a computationally-binding statistically-hiding bit-commitment yields statistical zero-knowledge arguments.

This allows us to limit our interest to bit-commitment protocols. Our focus is the assumptions required to construct such protocols. Quite a few constructions of bit-commitment protocol assuming different hardness assumption were presented. The situation is tight for computationally-hiding protocols: It was shown by Naor [Nao89] how to construct a perfectly-binding computationally-hiding bit-commitment using any "pseudorandom generator" which in turn can be constructed

[1] We remark that this notion may be preferable in settings where the zero-knowledge guarantee is critical. In such protocols the soundness is compromised only if the prover is unbounded *during* the execution of the protocol. It does not help the prover to "gain more computational power" after the execution. Nevertheless, the zero-knowledge guarantee holds "forever" even if the verifier's "gains more computational power" after the execution.

from any one-way function by Hastad et al. [HILL98]. On the other hand it was shown by [IL89] that bit-commitment implies the existence of one-way functions.

The picture is less clear for perfect/statistially-hiding bit-commitment. Perfectly-hiding computationally-binding bit-commitment protocols were constructed under: (1) specific algebraic assumptions [BMO90, BKK87, BCC88, BY90, IN93, IY87], (2) under the assumption that "collision intractable hash functions" exist [NY01] and (3) using one-way permutations [NOVY98]. However, it is not known whether such a protocol can be based only on the existence of one-way functions.

1.3 Our results

We construct a statistically-hiding computationally-binding bit-commitment (and hence construct a statistically-hiding zero-knowledge arguments) using the assumption that there exist approximable-preimage-size one-way functions:

Approximable-preimage-size one-way functions: are one-way functions with the extra feature that there is a feasible way to approximate the number of preimages of a given output of the function.

We remark that a special case are *regular one-way functions* in which every output element has the same number of preimages. This construction is an improvement to the protocol presented by [NOVY98] (which requires one-way permutations). [2] It can be viewed as a step towards narrowing the gap between the current implementation of statistically-hiding computationally-binding bit-commitment and the lower bound that states that statistically-hiding computationally-binding bit-commitment implies the existence of one-way functions [IL89].

1.4 Our Technique

Our protocol is based on the protocol of Naor et al. [NOVY98]. This protocol implements perfectly-hiding bit-commitment when applied using a one-way permutation. It is natural to ask what happens when this protocol is applied using a one-way function (rather than a permutation).

Given a one-way function $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ we distinguish between two distributions: $Y = f(U_n)$ (the distribution of the function when applied on a uniformly chosen input) and $Y' = U_{l(n)}$ (the uniform distribution over the outputs). Note that $Y = Y'$ for a one-way permutation.

We first observe that the main argument of [NOVY98] gives that for any function f, if the binding guarantee is broken then f can be inverted on the distribution Y'. (Note that this does not necessarily mean that f can be inverted on the distribution Y).

We also present a modified protocol (which preserves the binding guarantee above) and on which we can prove that if Y and Y' are "sufficiently similar" in a sense to be explained later, then the amount of information on the content of the envelope that is "leaked" during the execution can be bounded.

This motivates constructing one-way functions where the distribution Y is as "similar as possible" to Y'. A little bit more precisely, we show that if Y is a "dense distribution" meaning that (1) Y has "high" Renyi-entropy, and (2) Y does not assign "too low" probability to any element, then the modified protocol is computationally-binding and "somewhat statically hiding". Such a protocol can be amplified to give a computationally-binding and statistically-hiding protocol.

[2] We note that the protocol we achieve is only statistically-hiding whereas the protocol in [NOVY98] is perfectly-hiding.

The challenge is to transform an arbitrary one-way function into a one-way function with dense output distribution. We are not able to fulfill this task using any one-way function, and only succeed when given an approximable-preimage-size one-way function. For this task we use "hashing techniques" inspired by [HILL98]. Loosely speaking, by hashing $f(x)$ and x into smaller domains we can hope to get a more dense distribution. A concern is that this process might reveal information that will allow an adversary to invert the function.

1.5 Organization of the paper

In Section 2 we give notations and definitions used in this paper. In Section 3 we review the bit-commitment protocol presented by [NOVY98], "the NOVY protocol", and present a modified version of the protocol that yields stronger results. In Section 4 we are using one-way functions with approximable-preimage-size to construct one-way functions that are, in a sense, close to being permutations. In Section 5 we combine the results proven in the previous sections to achieve a computationally-binding statistically-hiding bit-commitment. Finally in Section 6 we conclude that the existence of non-uniform approximable-preimage-size one-way functions yields a statistical zero-knowledge arguments for every language $L \in NP$.

2 Preliminaries

- A function $\mu : \mathbb{N} \to [0,1]$ is *negligible* if for every positive polynomial $p(\cdot)$, $\mu(n) < 1/p(n)$ for large enough n.
- A function $\mu : \mathbb{N} \to [0,1]$ is *noticeable* if there exists a positive polynomial $p(\cdot)$ such that $\mu(n) > \frac{1}{p(n)}$ for large enough n.

2.1 Distributions and Entropy

We denote by U_n the uniform distribution over $\{0,1\}^n$. Given a function $f : \{0,1\}^n \to \{0,1\}^{l(n)}$, we denote by $f(U_n)$ the distribution over $\{0,1\}^{l(n)}$ induced by f operating on the uniform distribution. Given a distribution D over some set X, the support of D is defined as:

$$sup(D) = \{x \in X | D(x) > 0\}$$

Given two distribution X and Y, the statistical difference between them is defined as:

$$stat(X,Y) = \frac{1}{2} \sum_{z \in (sup(X) \bigcup sup(Y))} |X(z) - Y(z)|$$

Let D be a distribution over some finite domain X, we use the following "measures" of entropy:

- The min-entropy of D is $H_\infty(D) = \min_{x \in X} \log \frac{1}{D(x)}$.
- The collision-probability of D is $CP(D) = \sum_{x \in X} D(x)^2$.
- The Renyi entropy of D is $H_2(D) = \log \frac{1}{CP(D)}$.

2.2 Some technical Lemmas on probability distributions

Our proof requires several standard Lemmas on probability distributions. We include proofs for completeness.

Lemma 2.1 *Let D be a distribution over $\{0,1\}^n$ such that $CP(D) \leq \frac{1+\delta}{2^n}$, then $stat(D,U_n) \leq \frac{\sqrt{\delta}}{2}$.*

Proof: On one hand

$$\| D - U_n \|_2^2 = \| D \|_2^2 + \| U_n \|_2^2 - 2 < D, U_n > = CP(D) - \frac{1}{2^n} \leq \frac{\delta}{2^n}$$

On the other hand by the Chebyshev Sum Inequality

$$\| D - U_n \|_1^2 \leq 2^n \| D - U_n \|_2^2$$

Hence

$$stat(D,U_n) \stackrel{\text{def}}{=} \frac{1}{2} \| D - U_n \|_1 \leq \frac{\sqrt{\delta}}{2}$$

▮

Lemma 2.2 *Let D be a distribution over $\{0,1\}^n$ such that $H_2(D) \geq k$ then for every $\epsilon > 0$ there exists a distribution D' such that $H_\infty(D') \geq k - log(\frac{1}{\epsilon})$ and $stat(D,D') \leq \epsilon$.*

Proof: Let Z be a distribution defined over $[0,1]$ as $Z(y) = \sum_{x \in \{0,1\}^n, D(x)=y} y$, hence $E(Z) = CP(D) \leq 2^{-k}$. Therefore by the Markov inequality we have that for any $c > 0$,

$$Pr_Z[x \geq c2^{-k}] \leq \frac{1}{c}$$

Now let D' be the distribution obtained from D by "flattening" the probability of all the elements with probability higher than $c2^{-k}$, it is easy to see that $stat(D,D') \leq \frac{1}{c}$ and $H_\infty(D') \geq k - log(c)$. We are done by letting $\epsilon = \frac{1}{c}$. ▮

Lemma 2.3 *Let D and D' be distributions over $\{0,1\}^n$ and let ϵ and k be positive constants such that $stat(D,D') \leq \epsilon$ and $H_2(D') \geq k$, then there exists a set $B \subseteq \{0,1\}^n$ such that the following hold:*

- $Pr_{x \in_D \{0,1\}^n}[x \in B] \leq 4\epsilon$.
- $\forall y \notin B \; Pr_{x \in_D \{0,1\}^n}[x = y] \leq 2^{1-k}$.

Proof: Assume not, therefore there exists a set $B \subseteq \{0,1\}^n$ such that:

- $Pr_{x \in_D \{0,1\}^n}[x \in B] \geq 4\epsilon$.
- $\forall y \in B \; Pr_{x \in_D \{0,1\}^n}[x = y] > 2^{1-k}$.

Hence $\| D - D' \|_1 > 4\epsilon - \frac{4\epsilon}{2^{1-k}} \cdot 2^{-k} = 2\epsilon$ a contradiction. ▮

The following lemma is a combination of Lemmas 2.2 and 2.3:

Corollary 2.4 *Let D be a distribution over $\{0,1\}^n$ such that $H_2(D) \geq k$ and let ϵ be any positive constant, then there exists a set $B \subseteq \{0,1\}^n$ such that the following hold:*

- $Pr_{x \in_D \{0,1\}^n}[x \in B] \leq 4\epsilon$.
- $\forall y \notin B \; Pr_{x \in_D \{0,1\}^n}[x = y] \leq \frac{2^{1-k}}{\epsilon}$.

2.3 Universal hash functions

Definition 2.5 *Let $H_n : \{0,1\}^n \times \{0,1\}^{j(n)} \to \{0,1\}^{l(n)}$ be a collection of functions indexed by n such that $l(n) \leq n$. H_n is a universal family of hash functions if H_n, $j(n)$ and $l(n)$ are polynomial computable functions, and for all n, for all $x \in \{0,1\}^n$ and $x' \in \{0,1\}^n \setminus \{x\}$ and all $y, y' \in \{0,1\}^{l(n)}$,*

$$Pr_{j \in_R \{0,1\}^{j(n)}}[H_n(x,j) = y \text{ and } H_n(x',j) = y'] = \frac{1}{2^{2l(n)}}$$

Given $j \in \{0,1\}^{j(n)}$ we define $h_j : \{0,1\}^n \to \{0,1\}^{l(n)}$ as $h_j(x) = H_n(x', j)$.

There are few efficient constructions of universal hash functions for any values of n and $l(n)$ whose index length (i.e., $|j(n)|$) is polynomial or even linear in n [CW77, GL89].

2.4 Different types of one-way functions

We now define several types of one-way functions that come up in the paper.

Definition 2.6 (One-way function) *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is one-way if for every probabilistic-polynomial-time algorithm A, the function*

$$\mu(n) = Pr_{x \in \{0,1\}^n}[A(f(x)) \in f^{-1}(f(x))]$$

is negligible. A one-way permutation is a function f that is a permutation over any input length n.

Definition 2.7 (One-way function with approximable-preimage-size) *Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function, f has approximable-preimage-size if the function*

$$\tilde{D}_f(y) \stackrel{def}{=} \lceil \log(|f^{-1}(y)| \rceil$$

is polynomial computable [3]*.*

A particular case is that of a *regular* one-way function. This is a one-way function such that for any input length n, each output has the same number of preimages, and furthermore there is a polynomial-time algorithm which given n computes this value. We now define two special versions of one-way functions, with additional requirements on their output distribution.

Definition 2.8 (One-way function with high-output-entropy) *Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function, f has high-output-entropy if $H_2(f(U_n)) > n - 1$.*

Definition 2.9 (One-way function with δ-dense-output-distribution) *Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function, f has δ-dense-output-distribution if $H_2(f(U_n)) \geq l(n) - \delta$ and there exists a positive constant c such that $l(n) > n - c \cdot log(n)$.*

[3] We note that as far as the scope of this paper, we may allow the algorithm that computes $\tilde{D}_f$ to have additive errors of order $O(log(n))$ and to fail completely with some negligible probability.

2.5 Bit-commitment protocol

Definition 2.10 (Bit-commitment protocol) *A Bit-commitment is an interactive two-party protocol between a* `sender` *and a* `receiver`*. The protocol has two stages:*

- *The commit stage: the* `sender`*, $\mathcal{S}$, has a (secret) bit b to which he wishes to commit to the* `receiver`*, $\mathcal{R}$. $\mathcal{S}$ and $\mathcal{R}$ exchange message(s), where at the end of this stage $\mathcal{R}$ possessed a "commitment" to b without knowing its value.*
- *The reveal stage: $\mathcal{S}$ reveals b to $\mathcal{R}$ and try to persuade him that this is the bit he had committed to in the commit-stage. At the end of this stage $\mathcal{R}$ either accepts or rejects.*

The security of a bit-commitment protocol is measured by the following criterions:

Hiding: *Let $\mathcal{R}^*$ be some strategy for the receiver. Let $VIEW_{\mathcal{R}^*}(b)$ be the random variable defined from the $\mathcal{R}^*$'s view of the commit-stage of the protocol where $\mathcal{S}$ has committed to the bit b. Intuitively, the protocol is "hiding" if for every "allowed" strategy $\mathcal{R}^*$, $VIEW_{\mathcal{R}^*}(0)$ and $VIEW_{\mathcal{R}^*}(1)$ are "close". We distinguish between several variants:*

Perfectly-hiding: *For every strategy $\mathcal{R}^*$, $VIEW_{\mathcal{R}^*}(0) = VIEW_{\mathcal{R}^*}(1)$.*

Statistically-hiding: *For every strategy $\mathcal{R}^*$, $stat(VIEW_{\mathcal{R}^*}(0), VIEW_{\mathcal{R}^*}(1))$ is negligible. A protocol is* ϵ-Statistically-hiding *for some function $\epsilon(n)$ if $stat(VIEW_{\mathcal{R}^*}(0), VIEW_{\mathcal{R}^*}(1)) \leq \epsilon(n)$.*

Computationally hiding: *For every probabilistic-polynomial-time strategy $\mathcal{R}^*$, $VIEW_{\mathcal{R}^*}(0)$ and $VIEW_{\mathcal{R}^*}(1)$ are "computationally indistinguishable".*[4]

Binding: *Let $\mathcal{S}^*$ be some strategy for the sender. We say that $\mathcal{S}^*$ can* cheat *if following the commit-stage, he can persuade $\mathcal{R}$ to accept on both values of b. Intuitively, the protocol is "binding" if every "allowed" strategy $\mathcal{S}^*$ cannot cheat. We distinguish between several variants:*

Perfectly-binding: *No strategy $\mathcal{S}^*$ can cheat.*

Almost perfectly-binding: *No strategy $\mathcal{S}^*$ can cheat with more than negligible probability.*

Computationally-binding *No probabilistic-polynomial-time strategy $\mathcal{S}^*$ can cheat with more than negligible probability. A protocol is* ϵ-Computationally-binding *for some function $\epsilon(n)$ if no probabilistic-polynomial-time strategy $\mathcal{S}^*$ can cheat with probability larger than $\epsilon(n)$.*

Remark 2.11 (Uniform and non-uniform bit-commitment) *In the definition above we define "uniform versions" of bit-commitment. By that we mean that security is guaranteed against adversaries that are* uniform *(that is polynomial time probabilistic algorithms). One can analogously define "nonuniform versions" by replacing "polynomial time probabilistic algorithms" by "polynomial size circuits". This gives a stronger guarantee of security.*

This stronger notion is required when applying a computationally-binding bit-commitment into a perfect/statistical-zero-knowledge argument.

The construction of this paper (as well as all the constructions we are aware of) gives a uniform reduction that transforms an adversary that "breaks" the computational guarantee of the commitment scheme into an inverting procedure for the one-way function used. A consequence is that our

[4] That is, that every probabilistic-polynomial-time machine distinguishes between the two distributions with negligible probability, where the success probability is measured as a function of the security parameter of the protocol.

constructions and results immediately translate to the "nonuniform setting" and give a construction of non-uniform *computationally-binding statistically-hiding bit-commitment given a* non-uniform *approximable-preimage-size one-way function (that is one that cannot be inverted by polynomial size circuits). We use this observation to construct statistical zero-knowledge arguments in Section 6.*

3 The NOVY protocol and its consequences

An ingredient in our construction is the NOVY protocol for bit-commitment presented by Naor et al. [NOVY98]. The latter is a computationally-binding perfectly-hiding bit-commitment based on any one-way permutation. We now describe the protocol. For future use we give a general definition of the protocol where the function used by the protocol is any one-way function and not necessarily a one-way permutation.

Definition 3.1 *The NOVY protocol (using one-way function $f : \{0,1\}^n \to \{0,1\}^{l(n)}$) - Let $x \odot y$, where x and y are two vectors of the same length, be the inner-product of x and y modulo 2 and let $b \in \{0,1\}$ be the bit that $\mathcal{S}$ wants to commit to.*

Commit stage

C1. $\mathcal{S}$ selects a random $x \in \{0,1\}^n$ and computes $y = f(x)$. $\mathcal{S}$ keeps both x and y secret from $\mathcal{R}$.

C2. For j from 1 to $l(n) - 1$

- *(a) $\mathcal{R}$ selects a random $r_j \in \{0,1\}^{l(n)-j-1}$ and sends $h_j = 0^{j-1} \parallel 1 \parallel r_j$ to $\mathcal{S}$. (where $\parallel$ denotes the concatenation of the strings).*
- *(b) $\mathcal{S}$ sends $a_j = h_j \odot y$ to $\mathcal{R}$.*

C3. At this point there are exactly two vectors $y_0, y_1 \in \{0,1\}^{l(n)}$, such that for both $i \in \{0,1\}$, for all $1 \le j \le l(n) - 1$, $a_j = h_j \odot y_i$. Fix some ordering on $\{0,1\}^{l(n)}$ and assume $y_0 < y_1$. There exists $i \in \{0,1\}$ such that $y_i = y$. $\mathcal{S}$ sends $d = b \oplus i$ to $\mathcal{R}$.

Reveal stage

R1. $\mathcal{S}$ sends x and b to $\mathcal{R}$.

R2. $\mathcal{S}$ verifies that for all j in $[1 \ldots l(n) - 1]$, $a_j = h_j \odot y$ and that $f(x) = y_i$ where $i = b \oplus d$.

We now give intuition why this protocol works when f is a one-way permutation. From $\mathcal{R}$'s point of view it is equally likely that $y = y_0$ or $y = y_1$ and therefore $\mathcal{R}$ gets no information about b in the commit stage. On the other hand the fact that the h's are chosen at random prevents $\mathcal{S}$ from inverting y_{1-i} and thus he cannot cheat. The reason that the h's are given to $\mathcal{S}$ one-by-one (Line C2.), is to prevent a dishonest $\mathcal{S}$ from selecting x *after* seeing the h's, an advantage that could have enable $\mathcal{S}$ to find the preimages of both y's (and thus to cheat in the reveal stage).

Theorem 3.2 *([NOVY98, Theorem 2]) The above protocol when f is a one-way permutation is a computationally-binding perfectly-hiding bit-commitment protocol.*

We use (a variant of) the NOVY protocol with one-way functions f that are not permutations. It is helpful to abstract the following property of the protocol.

Lemma 3.3 *Assume that there exists a probabilistic polynomial time strategy $\mathcal{S}^*$ that following the commit-stage of the modified NOVY protocol, using $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ as its underlined one-way function, can reveal to an honest* **`receiver`** *two different values of b with non-negligible probability. Then there exists a probabilistic polynomial time algorithm A that inverts f on a* ***uniformly chosen*** *element in $\{0,1\}^{(l(n)}$ with non-negligible probability*

Proof: It is proven in [NOVY98, Lemma 2] that the existence of such S' implies the existence of a probabilistic polynomial time algorithm A that inverts f on a uniformly chosen element in $\{0,1\}^{l(n)}$ with non-negligible probability. This proof applies to any function f (although in [NOVY98] it is claimed only for a one-way permutation). ∎

Note that in Lemma 3.3 above the algorithm A inverts f with respect to the uniform distribution on the output, rather than the distribution of the function on a uniformly chosen input. These two distributions coincide when f is a one-way permutation. We show that these two distributions are "close enough" when f is a dense-one-way function.

3.1 A modified NOVY protocol

We now present a modified version of Protocol 3.1. In this version the receiver's selection of $h_1, \ldots, h_{l(n)-1}$ is done using a "coin-tossing" protocol which prevents a dishonest receiver from choosing $h_1, \ldots, h_{l(n)-1}$ maliciously as a function of the messages he receives.

Definition 3.4 *The modified NOVY protocol (the modified steps)*

C2.' For j from 1 to $l(n)-1$

- *(a) $\mathcal{R}$ selects a random $r_j \in \{0,1\}^{l(n)-j}$ and commits to the value of r_j through a perfect-binding computational-hiding protocol (recall that by [Nao89] such a protocol can be based on any one-way function).*
- *(b) $\mathcal{S}$ selects a random $w_j \in \{0,1\}^{l(n)-j}$ and sends w_j to $\mathcal{R}$.*
- *(c) $\mathcal{R}$ reveals the value of r_j to $\mathcal{S}$ and both parties set $h_j = 0^{j-1} \parallel 1 \parallel (r_j \oplus w_j)$.*
- *(d) $\mathcal{S}$ sends $a_j = h_j \odot y$ to $\mathcal{R}$.*

Before explaining the usefulness of the above modification for the hiding property of the protocol we observe that the binding property was not damaged.

Claim 3.5 *Let f be some one-way function. If the NOVY protocol using f is computationally-binding then the modified protocol is also computationally-binding.*

Proof: Every receiver strategy $\mathcal{R}^*$ for the modified protocol can also be executed in the original protocol. ∎

The advantage of using the modified version of the NOVY protocol is given in the following lemma.

Lemma 3.6 *Let f be a one-way function with δ-dense-output-distribution for $\delta \le (1/26)^2$. There exists some positive polynomial $p(n)$ such that The modified NOVY protocol based on f is $(1 - 1/p(n))$-statistically-hiding.*

Proof: Let $\mathcal{R}^*$ be some receiver strategy. We now examine the view of $\mathcal{R}^*$ where $\mathcal{S}$ has committed to a bit b. Throughout the proof we consider the probability space induced by the coin tosses of the two parties. In particular, x, y, $r_1, \ldots, r_{l(n)-1}$, $w_1, \ldots, w_{l(n)-1}$, $h_1, \ldots, h_{l(n)-1}$, $a_1, \ldots, a_{(l(n)-1}$, y_0, y_1, i and d are the random variables induced by this probability space. The modification in the protocol allows us to prove the following Claim.

Claim 3.7 *The random variables $h_1, \ldots, h_{l(n)-1}$ are independent of x, and are distributed identically no matter whether $b = 0$ or $b = 1$.*

Proof: Immediate from the protocol description. ∎

Note that the view of $\mathcal{R}^*$ contains the random variables $\bar{h} = (h_1, \ldots, h_{l(n)-1})$, $\bar{a} = (a_1, \ldots, a_{l(n)-1})$ and $d = b \oplus i$. Let h' and a' be some *fixed* values of $\bar{h}$ and $\bar{a}$ respectively. Let $E_{h',a'}$ be the event $\left\{\bar{h} = h'\right\} \cap \{\bar{a} = a'\}$. Note that the pair y_0, y_1 are fixed to some values y_0', y_1' in the event $E_{h',a'}$. We call y_0', y_1' the preimages of $E_{h',a'}$. Values h', a' are called ρ-hiding if

$$|\Pr[y = y_0'|E_{h',a'}] - \Pr[y = y_1'|E_{h',a'}]| \leq \rho$$

We call a pair y_0', y_1' t-balanced if

$$1/t \leq \frac{\Pr[y = y_0']}{\Pr[y = y_1']} \leq t$$

Claim 3.8 *Let h', a' be some values, and let y_0, y_1 be the associated pair of preimages. If y_0, y_1 are t-balanced then h', a' are $(1 - 1/2t)$-hiding.*

Proof: For every fixed values h' and a' we define the event $E'_{h',a'} = \cap_j \left\{h_j' \odot y_i = a_j'\right\}$. Note that $E'_{h',a'}$ is an event that is defined *only* over the choice of x made by the sender.

By Claim 3.7 we have that $\bar{h}$ is independent of x (and therefore independent of y). Thus, for every $b \in \{0,1\}$, and h', a',

$$\Pr[y = y_b|E_{h',a'}] = \Pr[y = y_b|E'_{h',a'}] = \Pr[y = y_b|y \in \left\{y_0', y_1'\right\}]$$

where y_0', y_1' is the pair associated with h' and a'. Note that

$$\frac{\Pr[y = y_0']}{\Pr[y = y_1']} = \frac{\Pr[y = y_0'|y \in \{y_0', y_1'\}]}{\Pr[y = y_1'|y \in \{y_0', y_1'\}]}$$

The claim now follows from the fact that y_0, y_1 are t-balanced by observing that for two numbers $p_0, p_1 \in (0, 1]$, if $1/t \leq \frac{p_1}{p_0} \leq t$ then $|p_1 - p_0| \leq 1 - 1/2t$.
∎

Note that for every h' and $y' \in \{0,1\}^{l(n)}$, we can associate a value a' (which is obtained when the sender uses y') and consequently a pair of preimages y_0', y_1'. For every value h' we define:

$$G_{h'} = \left\{y' : y_0', y_1' \text{ associated with } h' \text{ and } y' \text{ are } 2n^c\text{-balanced}\right\}$$

where c is the constant such that $l(n) \geq n - c \cdot \log(n)$ guaranteed from the assumption that f has a dense-output-distribution.

Claim 3.9 *For every h', $\Pr[f(x) \in G_{h'}] \geq 1 - 13\sqrt{\delta}$.*

Proof: We have that $H_2(f(U_n)) \geq l(n) - \delta$. We apply Lemma 2.1 and conclude that $stat(f(U_n), U_{l(n)}) \leq \sqrt{\delta}$. We apply Lemma 2.3 (with $k = n$) and conclude that there exists a set B_1 such that

- $Pr[f(x) \in B_1] \leq 4\sqrt{\delta}$.
- $\forall y' \notin B_1 \; Pr[f(x) = y'] \leq 2^{-l(n)+1}$.

Let $B_2 = \{0,1\}^{l(n)} \backslash sup(f(X))$. We have that $\Pr[f(x) \in B_2] = 0$ and therefore $\Pr_{y \in_R \{0,1\}^{l(n)}}[y \in B_2] \leq \sqrt{\delta}$.

¿From the assumption on f we also have that there exists a positive constant $c > 0$ such that $l(n) > n - c \cdot \log(n)$. This means that for every $y \notin B_2$, $\Pr[f(x) = y] \geq 2^{-n} \geq 2^{-l(n)}/n^c$.

Let B_3 be the set of all y' such that one of the preimages of y' using h' is in $B_1 \cup B_2$. It follows that $|B_3| \leq 2|B_1 \cup B_2|$. We want to show that $\Pr[f(x) \in B_3]$ is small.

$$\Pr[f(x) \in B_3] \leq \frac{|B_3|}{2^{l(n)}} + \sqrt{\delta} \leq 2 \cdot \left(\frac{|B_1|}{2^{l(n)}} + \frac{|B_2|}{2^{l(n)}} \right) + \sqrt{\delta} \leq 13\sqrt{\delta}$$

where the first equality holds because $stat(f(U_n), U_{l(n)}) \leq \sqrt{\delta}$ and the second holds because of the bound on the bound on the size of B_3. The third inequality holds by the bounds we have on the probabilities of B_1 and B_2.

For every $y' \notin B_3$, let y'_0, y'_1 be the associated preimages using h'. We have that for every $b \in \{0,1\}$,

$$\frac{2^{-l(n)}}{n^c} \leq \Pr[y = y'_b] \leq 2 \cdot 2^{-l(n)}$$

Thus, the pair is $2n^c$-balanced.

Let $G'_{h'} = \{0,1\}^{l(n)} \setminus B_3$. We conclude that for every $y' \in G'_{h'}$, y' and its other preimage are $2n^c$-balanced. Thus, $G'_{h'} \subseteq G_{h'}$ and the claim follows. ∎

To conclude the proof we need to bound $stat(VIEW_{\mathcal{R}^*}(0), VIEW_{\mathcal{R}^*}(1)$. Note that for every h', a' which are ρ-hiding, the distance between the two variables conditioned on $E_{h',a'}$ is at most ρ. This is because the only difference in the views under the conditioning is in the random variable c.

It follows from Claim 3.9 that:

$$\Pr[\bar{h}, \bar{a} \text{ are } 2n^c\text{-balanced}] \geq 1 - 13\sqrt{\delta}$$

and therefore using Claim 3.8

$$\Pr[\bar{h}, \bar{a} \text{ are } 1 - 1/4n^c\text{-hiding}] \geq 1 - 13\sqrt{\delta}$$

We conclude that the protocol is $(13\sqrt{\delta} + (1 - 13\sqrt{\delta})(1 - 1/4n^c))$-statistically-hiding. Note that $13\sqrt{\delta} + (1 - 13\sqrt{\delta})(1 - 1/4n^c) \leq 1 - 1/8n^c$ for $13\sqrt{\delta} \leq 1/2$ which is satisfied by our requirement on δ. ∎

4 One-way function with approximable-preimage-size to one-way function with dense-output-distribution

In this section we show how to transform any one-way function with approximable-preimage-size into a one-way function with dense output distribution.

Theorem 4.1 *If there exist one-way functions with approximable-preimage-size then for any $\delta > 0$ there exist one-way functions with δ-dense-output-distribution.* [5]

The construction is done through the following two steps:

4.1 One-way function with approximable-preimage-size to one-way function with high-output-entropy

The following construction appeared in the seminal paper of Hastad et al. [HILL98]: Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function with approximable-preimage-size, and let $H_n : \{0,1\}^n \times \{0,1\}^{j(n)} \to \{0,1\}^n$ be a universal family of hash functions. We define $g(x,j) : \{0,1\}^n \times \{0,1\}^{j(n)} \to \{0,1\}^{l(n)+n+j(n)}$ as:

$$g(x,j) = f(x) \parallel h_j(x)_{1\ldots(\tilde{D}_f(f(x))+2)} \parallel 0^{n-(\tilde{D}_f(f(x))+2)} \parallel j$$

where $h_j(x)_{1\ldots m}$ stands for the *first* m bits of $h_j(x)$.

The following claim, proven in [HILL98, Lemma 5.2], shows that g is a one-way function with high-output-entropy:

Claim 4.2 *g constructed above has the following properties:*

1. *g is a one-way function.*
2. *$H_2(g(U_n, U_{j(n)})) > n + j(n) - 1$.*

4.2 One-way function with high-output-entropy to one-way function with δ-dense-output-distribution

We assume for simplicity that $\delta < 1$. Let $f : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a one-way function with high-output-entropy, let $m = \lfloor n - 1 - log(\frac{1}{\delta}) \rfloor$ and let $H_n : \{0,1\}^{l(n)} \times \{0,1\}^{j(n)} \to \{0,1\}^m$ be a universal family of hash functions. We define $g : \{0,1\}^n \times \{0,1\}^{j(n)} \to \{0,1\}^{m+j(n)}$ as:

$$g(x,j) = h_j(f(x)) \parallel j$$

The following claim shows that g is a one-way function with δ-dense-output-distribution:

Claim 4.3 *g has the following properties:*

1. *g is a one-way function.*
2. *$H_2(g(U_n, U_{j(n)})) > m + j(n) - \delta$.*

Proving 4.3(2): The probability that two randomly chosen elements in $\{0,1\}^n$ have the same hash value is the sum of the following probabilities: The probability that the same element is chosen twice, which is the collision-probability of $f(U_n)$, and the probability that two *different* elements in $\{0,1\}^n$ has the same hash value, which equals by the property of a universal hash function $\frac{1}{2^m}$. Hence $CP(g(U_n, U_{j(n)})) = \frac{1}{2^{j(n)}}(CP(f(U_n) + \frac{1}{2^m}) < \frac{1}{2^{j(n)}}(\frac{2}{2^n} + \frac{1}{2^m}) \leq \frac{\delta+1}{2^{j(n)+m}}$ and thus $H_2(g(U_n, U_{j(n)})) \stackrel{\text{def}}{=} -\log(CP(g(U_n, U_{j(n)}))) > j(n) + m - \delta$ (where the last inequality holds since for any $\delta < 1$ $\log(1+\delta) < \delta$). ∎

[5] Actually, our construction works even for $\delta = o(1)$ as long as it is noticeable.

Proving 4.3(1): We assume towards a contradiction that g is not one-way and show that this leads to algorithm that inverts f with non-negligible success probability. By the contradiction assumption there exists a probabilistic polynomial time algorithm A, a polynomial fraction $\epsilon(\cdot)$ and infinity many n's such that:

$$Pr_{(x,j)\in_R\{0,1\}^{n+j(n)}}[A(g(x,h)) \in g^{-1}(x,j)] > \epsilon(n) \tag{1}$$

Using the second part of the current claim together with Corollary 2.4 (with $\epsilon = \frac{\epsilon(n)}{8}$) we have that there exists a set $B \subseteq \{0,1\}^{m+j(n)}$ such that:

$$\begin{aligned} &Pr_{(x,j)\in_R\{0,1\}^{n+j(n)}}[g(x,j) \in B] \leq \frac{\epsilon(n)}{2} \\ &\forall y \notin B \; Pr_{(x,j)\in_R\{0,1\}^{n+j(n)}}[g(x,j) = y] \leq \frac{32}{\epsilon(n)2^{m+j(n)}} \end{aligned} \tag{2}$$

It is easy to see that, since B is small, algorithm A works quite well even when restricted to inputs outside of B, that is:

$$Pr_{(x,j)\in_R(\{0,1\}^{n+j(n)}\setminus\, g^{-1}(B))}[A(g(x,j)) \in g^{-1}(x,j)] > \frac{\epsilon(n)}{2} \tag{3}$$

We now define algorithm M^A that inverts f with non-negligible success probability. Given $y \in \{0,1\}^{l(n)}$, algorithm M^A selects a random $j \in \{0,1\}^{j(n)}$ and returns $A(h(y) \parallel j)_1$ (i.e., the first element of $A(h(y) \parallel j)$). Note that even when A is successful in inverting $h_j(f(x)) \parallel j$ we are not guaranteed that M^A's answer is in $f^{-1}(x)$ (since its answer might be another $x' \in \{0,1\}^n$ such that $f(x') \neq f(x)$ but $h_j(f(x)) = h_j(f(x'))$. Fortunately, we are guaranteed that when restricting our interest to inputs outside of B, given that A is successful in inverting $(h_j(f(x)) \parallel j)$ implies that M^A's answer is in $f^{-1}(x)$ with noticeable probability. The proof follows:

$Pr[M^A(f(x)) \in f^{-1}(x)] \geq$ (ignoring the cases when $g(x) \in B$)

$Pr[M^A(f(x)) \in f^{-1}(x) \mid g(x) \notin B] \cdot Pr[g(x) \notin B] \geq$

(ignoring the cases when A is not successful in inverting $g(x)$)

$Pr[A(g(x))_1 \in f^{-1}(x) \mid g(x) \notin B \text{ and } A(g(x)) \in g^{-1}(g(x))]$

$\cdot\, Pr[A(g(x)) \in g^{-1}(g(x)) \mid g(x) \notin B] \cdot \Pr[g(x) \notin B] \geq$

(calculating the conditional probability that $A(g(x))_1 \in f^{-1}(x)$)

$\frac{2^{-(n+j(n))}}{\max_{x\notin g^{-1}(B)}(Pr[g(x)])} \cdot \left(\frac{\epsilon(n)}{2}\right)^2 \geq$ (by Equation 2)

$2^{-(n+j(n))} \cdot \frac{\epsilon(n)2^{m+j(n)}}{32} \cdot \left(\frac{\epsilon(n)}{2}\right)^2 \geq$ (by the definition of m)

$\frac{\epsilon(n)^3}{256}\frac{\delta}{2} = \frac{\epsilon(n)^3\delta}{512}$

∎

5 Computationally-binding statistically-hiding bit-commitment using one-way function with dense-output-distribution

In this section we put things together and prove our main theorem.

Theorem 5.1 (main theorem) *If there exist one-way functions with approximable-preimage-size then there are computationally-binding statistically-hiding bit-commitment protocols.*

Given a one-way function with approximable-preimage-size, we first apply Theorem 4.1 to convert it into a δ-one-way function for $\delta = (1/26)^2$. We now use the modified NOVY protocol from Section 3 with this function. By Lemma 3.6 we have that there exists some positive polynomial $p(n)$ such that this protocol is $(1-1/p(n))$-statistically-hiding. To show that it is computationally-binding we combine Lemma 3.3, Claim 3.5, and the following Lemma:

Lemma 5.2 *For every $\delta < 1$, let $f : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ be a one-way function with δ-dense-output-distribution then f is a one-way even with respect to the uniform distribution over its outputs.*

Proof: The proof is done by contradiction. We assume the exitance of an efficient algorithm A that inverts f with non-negligible probability over its outputs and show that it leads to the existence of an efficient algorithm that inverts f with non-negligible probability *over its inputs* (and thus contradicts the onewayness of f). Let $\epsilon(\cdot)$ be a polynomial fraction such that for infinity many n's:

$$Pr_{y \in_R \{0,1\}^{l(n)}}[A(y) \in f^{-1}(y)] > \epsilon(n)$$

where the probability in is also over the random coins of A. It follows that:

$$Pr_{y \in_R sup(f(U_n))}[A(y) \in f^{-1}(y)] > \epsilon(n)$$

By an averaging argument we have that there exists a set $Y \subseteq sup(f(U_n))$ such that:

$$\frac{|Y|}{sup(f(U_n)} \geq \frac{\epsilon(n)}{2} \tag{4}$$

$$\text{for all } y \in Y \; Pr[A(y) \in f^{-1}(y)] \geq \frac{\epsilon(n)}{2}$$

We have that $H_2(f(U_n)) \geq l(n) - \delta$. We apply Lemma 2.1 and conclude that $stat(f(U_n), U_{l(n)}) \leq \sqrt{\delta}$ and thus our requirement on δ implies that

$$sup(f(U_n) \geq 2^{l(n)-1}$$

Let c is the constant such that $l(n) > n - c \cdot \log(n)$ guaranteed from the assumption that f has a dense-output-distribution. Hence,

$$Pr_{x \in_R \{0,1\}^n}[f(x) \in Y] \geq \frac{|Y|}{2^n} \geq \frac{\epsilon(n)}{4n^c}$$

Therefore,

$$Pr_{x \in_R \{0,1\}^n}[A(f(x)) \in f^{-1}(f(x))] \geq Pr[A(f(x)) \in f^{-1}(f(x)) | f(x) \in Y] \cdot Pr[f(x) \in Y] \geq \frac{\epsilon(n)}{2}\frac{\epsilon(n)}{4n^c} = \frac{\epsilon(n)^2}{8n^c}$$

∎

Finally, in the following section we show how to amplify the binding property of the protocol and obtain a computationally-binding statistically-hiding commitment, thus completing the proof of Theorem 5.1.

5.1 Computationally-binding $(1-1/poly)$-statistically-hiding bit-commitment to computationally-binding statistically-hiding

Given a computationally-binding $(1-1/p(n))$-statistically-hiding bit-commitment. The two parties invoke the following protocol:

Commit stage

C1. $\mathcal{S}$ chooses $np(n)-1$ random bits $b_1, \dots b_{np(n)-1} \in \{0,1\}$ and sets $b_{np(n)} = b \oplus (\bigoplus_{j=1}^{np(n)-1} b_j)$.

C2. The two parties invoke the commit-stage of P for each of the bits.

Reveal stage

R1. The two parties invoke the reveal-stage of P for each of the b_i's.

R2. $\mathcal{R}$ sets $\bigoplus_{j=1}^{np(n)} b_j$ as the revealed value.

Claim 5.3 *The above protocol is a computationally-binding statistically-hiding bit-commitment protocol.*

Proof: The binding is by the straight forward hybrid argument relaying of the computationally-binding property of the underlined protocol. For the hiding property we are using the following claim, which is an immediate extension of [SV97, Prop. 3.6], to get that the protocol is a e^{-n}-statistically-hiding.

Claim 5.4 *Let $\{X_0^1, X_0^2, \dots, X_0^n\}$ and $\{X_1^1, X_1^2, \dots, X_1^n\}$ be two sequences of independent random variables, and let Y_k, for both $k \in \{0,1\}$, be the following random variable:*
Y_k: Choose $m_1, m_2, \dots, m_n$ uniformly in $\{0,1\}$ such that $\left(\bigoplus_{j=1}^{n} m_j\right) = k$. Output a sample of $(X_{m_1}^1, X_{m_2}^2, \dots, X_{m_n}^n)$.

Then

$$stat(Y_0, Y_1) = \prod_{j=1}^{n} stat(X_0^j, X_1^j)$$

■

6 Obtaining statistical zero-knowledge arguments

We now turn our attention to constructing statistical zero-knowledge arguments. A technicality, is that this requires a nonuniform version of bit-commitment. In Theorem 5.1 we have shown how to base a computationally-binding statistically-hiding bit-commitment on any approximable-preimage-size one-way function. The same result holds in the "non-uniform setting" (see the discussion in remark 2.11). That is, given an approximable-preimage-size function that is one-way against polynomial size circuits, we can construct a bit-commitment scheme that is computationally-binding against polynomial size circuits. We refer to both these notions as *non-uniform* in the results below.

The construction of the statistical zero-knowledge arguments is achieved by the above and the following theorem [6]:

[6] The actual theorem was stated for the perfectly-hiding case, however, the proof for the statistically-hiding case is essentially the same.

Theorem 6.1 *([BCC88]) If non-uniform computationally-binding statistically-hiding bit-commitment exists, then every language $L \in NP$ has a statistical zero-knowledge arguments.*

To conclude we have the following corollary:

Corollary 6.2 *If any non-uniform approximable-preimage-size one-way function exist, then every language $L \in NP$ has a statistical zero-knowledge arguments.*

7 Open problems

The natural question to ask is whether it is possible to construct a computationally-binding statistically-hiding bit-commitment using *any* one-way function. An alternative task would be to present a black box separation between these two primitives.

While we are not sure which of the two directions is more promising, we remark that using similar ideas to the ones used in this paper, it is possible to construct computationally-binding statistically-hiding bit-commitment in the random oracle model. This gives evidence that it may be hard to provide a black-box separation.
Another open problem is to reduce the number of rounds of communication in the NOVY protocol.

Acknowledgements

We are grateful to Oded Goldreich, Danny Harnik, Omer Reingold and Alon Rosen for helpful conversations.

References

[BCC88] Gilles Brassard, David Chaum, and Crepeau Crepeau. Minimum disclosure proofs of knowledge. *JCSS*, 37(2):156–189, October 1988.

[BKK87] Joan F. Boyar, Mark W. Krentel, and Kurtz Kurtz. A discrete logarithm implementation of zero-knowledge blobs. Technical Report TR-87-02, Department of Computer Science, University of Chicago, March 1987. Thu, 30 Nov 1995 22:05:46 GMT.

[BMO90] M. Bellare, S. Micali, and R. Ostrovsky. The (true) complexity of statistical zero knowledge. In Baruch Awerbuch, editor, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pages 494–502, Baltimore, MY, May 1990. ACM Press.

[BY90] Gilles Brassard and Moti Yung. One-way group actions. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology—CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 94–107. Springer-Verlag, 1991, 11–15 August 1990.

[CW77] I. L. Carter and M. N. Wegman. Universal classes of hash functions. In *Conference Record of the Ninth Annual ACM Symposium on Theory of Computing: Papers Presented at the Symposium, Boulder, Colorado, May 2–4, 1977*, pages 106–112, New York, NY 10036, USA, 1977. ACM Press.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In ACM, editor, *Proceedings of the twenty-first annual ACM Symp. on Theory of Computing,* *Seattle, Washington, May 15–17, 1989*, pages 25–32, New York, NY, USA, 1989. ACM Press.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive systems. *SIAM Journal of Computing*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, July 1991.

[Gol02] Oded Goldreich. Zero-knowledge. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS-02)*, pages 3–8, Los Alamitos, November 16–19 2002. IEEE COMPUTER SOCIETY.

[HILL98] Hastad, Impagliazzo, Levin, and Luby. A pseudorandom generator from any one-way function. *SICOMP: SIAM Journal on Computing*, 28, 1998.

[IL89] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In IEEE, editor, *30th annual Symposium on Foundations of Computer Science, October 30–November 1, 1989, Research Triangle Park, North Carolina*, pages 230–235, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1989. IEEE Computer Society Press.

[IN93] G. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. Technical report, July 01 1993.

[IY87] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations (extended abstract). In Carl Pomerance, editor, *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer-Verlag, 1988, 16–20 August 1987.

[Nao89] Naor. Bit commitment using pseudo-randomness. In *CRYPTO: Proceedings of Crypto*, 1989.

[NOVY98] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for np using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. preliminary version in CRYPTO 92.

[NY01] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications, February 18 2001.

[SV97] Amit Sahai and Salil Vadhan. A complete promise problem for statistical zero-knowledge. In *Proceedings of the 38th Annual Symp. on the Foundations of Computer Science*, pages 448–457. IEEE, October 1997.