Statistical Zero-Knowledge Arguments for NP from Any One-Way Function*

(Extended Abstract)

Minh-Huyen Nguyen Shien Jin Ong Salil Vadhan

Division of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts, USA.
E-mail: {mnguyen,shienjin,salil}@eecs.harvard.edu

Abstract

We show that every language in $\mathbf{NP}$ *has a* statistical *zero-knowledge argument system under the (minimal) complexity assumption that one-way functions exist. In such protocols, even a computationally unbounded verifier cannot learn anything other than the fact that the assertion being proven is true, whereas a polynomial-time prover cannot convince the verifier to accept a false assertion except with negligible probability. This resolves an open question posed by Naor, Ostrovsky, Venkatesan, and Yung (CRYPTO '92, J. Cryptology '98).*

Departing from previous works on this problem, we do not construct standard statistically hiding commitments from any one-way function. Instead, we construct a relaxed variant of commitment schemes called "1-out-of-2-binding commitments," recently introduced by Nguyen and Vadhan (STOC '06).

1 Introduction

As first discovered by Shannon [Sha] for the case of encryption, most interesting cryptographic tasks are impossible to achieve with absolute, information-theoretic security. Thus, modern cryptography aims to design protocols that are computationally intractable to break. Specifically, following Diffie and Hellman [DH], this is typically done by showing that breaking the protocol is as hard as some intractable problem from complexity theory. Unfortunately, proving lower bounds of the sort needed seems beyond the reach of current techniques in complexity theory, and indeed would require at least proving $\mathbf{P} \neq \mathbf{NP}$.

Given this state of affairs, research in the foundations of cryptography has aimed to design cryptographic protocols based on complexity assumptions that are as weak and general as possible. This project was enormously successful in the 1980's. In a beautiful sequence of works, it was shown that many cryptographic primitives, such as pseudorandom generators, pseudorandom functions, private-key encryption and authentication, digital signatures, (computationally hiding) bit commitment, and (computational) zero-knowledge proofs could be constructed from any one-way function [HILL, GGM, Rom, Nao, GMW], and moreover this complexity assumption is minimal in the sense that each of these primitives (and indeed almost any cryptographic task) implies the existence of one-way functions [IL, OW]. Moreover, it was shown that many of the remaining primitives, such as public-key encryption, collision-resistant hashing, and oblivious transfer, could not be reduced to the existence of one-way functions in a "black-box" manner [IR, Sim].

However, a few important primitives have resisted classification into the above categories. That is, it is only known how to build these primitives from seemingly stronger assumptions than the existence of one-way functions, yet there is no black-box separation between these primitives and one-way functions. In this work, we are interested in an example involving zero-knowledge protocols.

1.1 The Complexity of Zero Knowledge

Zero-knowledge proofs are protocols whereby one party, the *prover*, convinces another party, the *verifier*, that some assertion is true with the remarkable property that the verifier "learns nothing" other than the fact that the assertion being proven is true. Since their introduction by Goldwasser, Micali, and Rackoff [GMR1], zero-knowledge proofs have played a central role in the design and study of crypto-

* A full version of this paper is available on ECCC [NOV]. All three authors were supported by NSF grant CNS-0430336 and ONR grant N00014-04-1-0478.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

graphic protocols. Part of the reason for their vast applicability is the fact that, under certain complexity assumptions (discussed below), every language L in $\mathbf{NP}$ has a zero-knowledge proof system [GMW]. That is, a prover can efficiently convince a verifier that $x \in L$ in a zero-knowledge manner, provided that the prover possesses an $\mathbf{NP}$ witness to the membership of x in L. This means that when designing cryptographic protocols, any time that one party needs to convince others of some fact (e.g., that it has followed the specified protocol) without revealing additional knowledge, it can do so provided that it possesses a witness to the fact (e.g., its own secret keys and coin tosses).

Zero-knowledge protocols come in several flavors, depending on how one formulates the two security conditions: (1) the zero-knowledge condition, which says that the verifier "learns nothing" other than the fact the assertion being proven is true, and (2) the soundness conditions, which says that the prover cannot convince the verifier of a false assertion. In *statistical zero knowledge*, the zero-knowledge condition holds regardless of the computational resources the verifier invests into trying to learn something from the interaction. In *computational zero knowledge*, we only require that a probabilistic polynomial-time verifier learn nothing from the interaction.[1] Similarly, for soundness, we have *statistical soundness*, a.k.a. *proof systems*, where even a computationally unbounded prover cannot convince the verifier of a false statement (except with negligible probability), and *computational soundness*, a.k.a. *argument systems* [BCC], where we only require that a polynomial-time prover cannot convince the verifier of a false statement.

Of course, it would be ideal to have both security conditions be statistical, and thus hold against computationally unbounded adversaries. Unfortunately, the resulting notion, *statistical zero-knowledge proofs*, while quite interesting and nontrivial (cf., [Vad]), can only be achieved for languages in $\mathbf{AM}\cap\mathbf{coAM}$ [For, AH]. Because $\mathbf{AM}\cap\mathbf{coAM}$ is not believed to contain $\mathbf{NP}$ (cf., [BHZ]), it is unlikely that every problem in $\mathbf{NP}$ possess statistical zero-knowledge proofs. Thus at best, we can have one of the security conditions be statistical.

Computational zero-knowledge *proofs* (with statistical soundness) was the original notion proposed in [GMR1]. Goldreich, Micali, and Wigderson [GMW] showed that we can construct such proof systems for all of $\mathbf{NP}$ from any bit-commitment scheme that is computationally hiding and statistically binding. By [Nao, HILL], such commitment schemes can be constructed from any one-way function, and thus we obtain computational zero-knowledge proofs for $\mathbf{NP}$ from any one-way function. This complexity assumption is essentially minimal due to results of Ostrovsky and Wigderson [OW], who showed that zero-knowledge proofs for any non-trivial language imply a weak form of one-way functions.

Brassard, Chaum, and Crepeau [BCC] proposed instead the notion of *statistical zero-knowledge arguments*,[2] between the which is what we study in this paper. One reason that this variant of zero-knowledge proofs may be preferable to the original one is that breaking the soundness property must be done "on-line" during the interaction with the verifier and thus we need only protect against the adversary's present-day computational resources, whereas breaking the zero-knowledge property can involve the adversary investing effort long after the interaction to try and learn something from the transcript of the interaction. Thus, it seems preferable for the zero-knowledge property to be the one with the stronger, statistical guarantee.

It is evident from the constructions of [GMW, BCC] that to construct statistical zero-knowledge arguments for all of $\mathbf{NP}$, it suffices to construct bit-commitment schemes that are statistically hiding and computationally binding. The early constructions of such schemes were based on specific number-theoretic complexity assumptions [BCC, BKK], and were later generalized to any family of claw-free permutations [GK], and then to any family of collision-resistant hash functions [NY] (see also [DPP]).[3]

In 1992, Naor, Ostrovsky, Venkatesan, and Yung [NOVY] showed that the collision resistance criterion[4] is not necessary, by giving a beautiful construction of statistically hiding commitments (in fact perfectly hiding ones) and thus statistical zero-knowledge arguments for $\mathbf{NP}$ from any one-way *permutation*. They left as an open question whether these primitives could be based on arbitrary one-way functions, which would again be essentially minimal by [Ost, OW].[5]

The only progress in the past decade came in 2005 when Haitner et al. [HHK$^+$] showed how to construct statistically hiding commitments from any "approximable preimage size" one-way function, which is a one-way function where we can efficiently approximate the preimage size of points in the range.

[1] More precisely, in statistical zero knowledge, we require that the verifier's view of the interaction can be efficiently simulated up to negligible statistical distance, whereas in computational zero knowledge, we only require that the simulation be computationally indistinguishable from the verifier's view.

[2] Actually, [BCC] and some subsequent works (such as [NOVY]) constructed *perfect* zero-knowledge arguments, which intuitively guarantee that the verifier learns something from the interaction with *zero* probability (as opposed to negligible probability, as in statistical zero knowledge). However, this distinction is minor in comparison to the distinction between the statistical and computational zero knowledge, which refer to computationally unbounded and polynomial-time verifier strategies, respectively.

[3] The fact that claw-free permutations imply collision-resistant hash functions was shown in [GMR2, Dam], and the early constructions of claw-free permutations based on specific number-theoretic complexity assumptions were given by [GMR2, BKK].

[4] We note that one-way permutations and collision-resistant hashing are known to be incomparable under "black-box reductions" [Sim, Rud, KSS].

[5] The results of [Ost, OW] are stated only for proof systems, but they also hold for argument systems.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

Motivated by this recent development, in this paper we resolve the complexity of statistical zero-knowledge arguments for $\mathbf{NP}$:

Theorem 1.1. *If one-way functions*[6] *exist, then every language in* $\mathbf{NP}$ *has a statistical zero-knowledge argument system.*

Deviating from prior works on this problem, we do *not* prove this theorem by constructing the standard notion of statistically hiding commitments from any one-way function. Instead, as described below, we work with a relaxed variant of commitment schemes recently introduced by Nguyen and Vadhan [NV], which we describe in the next section.

We also remark that our protocol has a polynomial number of rounds, while a constant number of rounds can be achieved based on collision-resistant hashing [NY, BCY]. However, achieving a subpolynomial ($n^{o(1)}$) number of rounds is open even assuming the existence of one-way permutations (cf., [NOVY]).

1.2 Techniques

We begin by recalling the notion of a *commitment scheme*. A commitment scheme is a two-stage protocol between a sender and a receiver. In the first stage, the sender 'commits' to a value v, and in the second, the sender 'reveals' this value to the receiver. We want two security properties from a commitment scheme. The *hiding* property says that the receiver does not learn anything about the value v during the commit stage. The *binding* property says that after the commit stage, there is at most one value that the sender can successfully open (without the receiver rejecting). As with zero-knowledge protocols, each of these security properties can be computational or statistical. For commitments, it is impossible to have both properties be statistical. As mentioned earlier, statistically binding commitments can be constructed from any one-way function [Nao, HILL], but our interest is in statistically hiding commitments.

Recently, Nguyen and Vadhan [NV] introduced a new relaxation of commitment schemes, called *1-out-of-2-binding commitment schemes*, symbolically written as $\binom{2}{1}$-binding commitment schemes. These are commitment schemes with two phases, each consisting of a commit stage and a reveal stage. In the first phase, the sender commits to and reveals one value v_1, and subsequently, in the second phase, the sender commits to and reveals a second value v_2. We require that both phases are hiding, but only that one of them is binding. That is, the binding property only requires that with high probability, the sender will be forced to reveal the correct committed value in at least one of the phases (but which of the two phases can be determined dynamically by the malicious sender).

In [NV], it was shown that such commitment schemes still suffice to construct zero-knowledge protocols for all of $\mathbf{NP}$. Thus, our task is reduced to constructing $\binom{2}{1}$-binding commitment schemes that are statistically hiding and computationally binding from any one-way function. Unfortunately, we do not know how to do this. Instead, we construct polynomially many two-phase commitment schemes, with the guarantee that *at least one* of the schemes is hiding (in both phases), and *all* of the schemes are $\binom{2}{1}$-binding. Fortunately, a similar issue arose also in [NV] and it was shown how even such a collection could be used to construct zero-knowledge protocols for $\mathbf{NP}$.

Even though we draw upon [NV] for the notion of $\binom{2}{1}$-binding commitments and its utility for zero knowledge, there are many differences between the contexts of the two works and the constructions of $\binom{2}{1}$-binding commitments. In [NV], the goal was to prove *unconditional* results about prover efficiency in zero-knowledge proofs (that one can transform zero-knowledge proofs with inefficient provers into ones with efficient provers). This was done by showing that every problem having a zero-knowledge proof has an "instance-dependent" $\binom{2}{1}$-binding commitment scheme, where the sender and receiver get an instance x of the problem as auxiliary input and we only require hiding to hold when x is a "yes instance" and binding when x is a "no instance." Here, we are giving *conditional* results (assuming the existence of one-way functions) and are obtaining standard (as opposed to instance-dependent) $\binom{2}{1}$-binding commitments. Moreover, the focus in [NV] is on proof systems and *statistically* $\binom{2}{1}$-binding commitments; thus here we need to develop new formulations to work with argument systems and the computational binding property.

Our initial construction, which gives a $\binom{2}{1}$-binding commitment scheme satisfying a "weak hiding" property, is inspired by the construction of [NV]. Indeed, the second phase in [NV] was also introduced to deal with non-regular functions (corresponding to "non-flat distributions" in their setting), and our construction can be seen as applying the same idea to a variant of the protocol of [HHK$^+$]. However, in [NV], this construction immediately gives a "strong hiding" property, whereas much of the technical work in the current paper comes from amplifying the "weak hiding" property we obtain into a strong one.

2 Preliminaries

Let X be a *random variable* taking values in a finite set T. We write $x \leftarrow X$ to indicate that x is selected accord-

[6] As in most treatments of zero knowledge, we use a nonuniform notion of security, and thus require our one-way functions to be secure against nonuniform algorithms (i.e., circuits). Uniform treatments of zero-knowledge proofs and arguments are possible (see [Gol1, BLV]) but are much more cumbersome.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

ing to X. For a finite set S, we write $x \leftarrow S$ to indicate that x is selected uniformly amongst all the elements of S. The *support* of a random variable X is $\mathrm{Supp}(X) = \{x : \Pr[X = x] > 0\}$. Random variable X is *flat* if it is uniform over its support.

A negligible function, denoted by neg, is a function that vanishes more quickly than any inverse polynomial. That is, for all $c \in \mathbb{N}$, $\mathrm{neg}(n) < n^{-c}$ for all sufficiently large n. Let $\mathrm{poly}(n)$ denote any polynomial, that is $\mathrm{poly}(n) \leq n^c$ for some $c \in \mathbb{N}$, and for all sufficiently large n.

The *statistical difference* between two random variables A and B over $\{0,1\}^n$ is defined as $\Delta(A,B) \overset{\text{def}}{=} \max_{T \subseteq \{0,1\}^n} |\Pr[A \in T] - \Pr[B \in T]| = \frac{1}{2}\sum_{x \in \{0,1\}^n} |\Pr[A \in T] - \Pr[B \in T]|$. We say that distributions A and B are *ε-close* if $\Delta(A,B) \leq \varepsilon$.

Let I be a set of strings. A *probability ensemble* of a sequence of random variables indexed by I is denoted as $\{A_x\}_{x \in I}$. We say that two ensembles $\{A_x\}_{x \in I}$ and $\{B_x\}_{x \in I}$ are *statistically indistinguishable* if there exists a negligible function ε such that X_x and Y_x are $\varepsilon(|x|)$-close for every $x \in I$. We write $\{A_x\}_{x \in I} \approx_s \{B_x\}_{x \in I}$ to denote that the two ensembles are statistically indistinguishable.

For a probabilistic algorithm A, we write $A(x;r)$ to denote the output of A on input x and coin tosses r. PPT refers to probabilistic algorithms that run in *strict* polynomial time. A *nonuniform* PPT algorithm is a pair $(A, \bar{z})$, where $\bar{z} = z_1, z_2, \ldots$ is an infinite series of strings where $|z_n| = \mathrm{poly}(n)$, and A is a PPT algorithm that receives pairs of input of the form $(x, z_{|x|})$. (The string z_n is called the *advice string* for A for inputs of length n.) Nonuniform PPT algorithms are equivalent to families of polynomial-sized Boolean circuits.

Definition 2.1 (one-way function)**.** Let $s \colon \mathbb{N} \to \mathbb{N}$ be any function. A function $f \colon \{0,1\}^* \to \{0,1\}^*$ is a *$s(n)$-secure one-way function* if f is computable in polynomial time and for every nonuniform PPT A,

$$\Pr_{x \leftarrow \{0,1\}^n}[A(1^n, f(x)) \in f^{-1}(f(x))] < 1/s(n),$$

for all sufficiently large n. We say that f is a *one-way function* if f is $s(n)$-secure for every polynomial s.

We say that a one-way function f is *regular* with *preimage size $g(n)$* if there exists a function $g \colon \mathbb{N} \to \mathbb{N}$ such that $\forall y \in \mathrm{Supp}(f(U_n))$, $|\{x \in \{0,1\}^n : f(x) = y\}| = g(n)$.

2.1 Statistical Zero-Knowledge Arguments

We follow the standard definitions of *zero-knowledge arguments*, as in [Gol2, Sec. 4.8]. Roughly speaking, an *argument (or computationally sound proof system)* is an interactive protocol (P, V) whose soundness only holds against computationally bounded adversaries. That is for a language L, if $x \notin L$ then every nonuniform PPT adversarial prover P^* convinces V to accept with probability at most $1/3$. We define an interactive proof system being *statistical zero knowledge* as follows.

Definition 2.2 (statistical zero knowledge)**.** We say an argument system (P, V) is (black-box) *statistical zero knowledge* if there exists a universal PPT simulator S such that for all verifiers V^*, we have

$$\{\mathsf{view}_{V^*}(P, V^*)(x)\}_{x \in L} \approx_s \{S^{V^*}(x)\}_{x \in L},$$

where view_{V^*} denotes the *view* of V^*, which consists of the transcript of the interaction together with $V*$'s coin tosses.

The above definition of zero knowledge is a *black box* definition in the sense that the simulator is universal for all (even computationally unbounded) verifier strategies V^*, and in particular does not depend on the code of V^*. The zero-knowledge protocols we construct will all be black-box zero knowledge and thus satisfy the above definition.

2.2 1-out-of-2-Binding Commitments

We now introduce the notion of $\binom{2}{1}$-binding commitments that will play a central role in establishing our results. These are commitment schemes with two *sequential* and *related* stages such that in each stage, the sender commits to and reveals a value.

Definition 2.3. A *2-phase commitment scheme* (S, R), with security parameter n and message length $k = k(n)$, consists of four interactive protocols: (S_c^1, R_c^1) the first commitment stage, (S_r^1, R_r^1) the first reveal stage, (S_c^2, R_c^2) the second commitment stage, and (S_r^2, R_r^2) the second reveal stage. For us, both reveal phases will always be noninteractive, consisting of a single message from the sender to the receiver. Throughout, both parties receive the security parameter 1^n as input.

1. In the first commitment stage, S_c^1 receives a private input $\sigma^{(1)} \in \{0,1\}^k$ and a sequence of coin tosses r_S. At the end, S_c^1 and R_c^1 receive as common output a commitment $z^{(1)}$. (Without loss of generality, we can assume that $z^{(1)}$ is the transcript of the first commitment stage.)

2. In the first reveal stage, S_r^1 and R_r^1 receive as common input the commitment $z^{(1)}$ and a string $\sigma^{(1)} \in \{0,1\}^k$ and S_r^1 receives as private input r_S. At the end, S_r^1 and R_r^1 receive a common output τ. (Without loss of generality, we can assume that τ is the transcript of the first commitment stage and the first reveal stage and includes R_r^1's decision to accept or reject.)

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

3. In the second commitment stage, S_c^2 and R_c^2 both receive the common input $\tau \in \{0,1\}^*$, and S_c^2 receives a private input $\sigma^{(2)} \in \{0,1\}^k$ and the coin tosses r_S. S_c^2 and R_c^2 receive as common output a commitment $z^{(2)}$. (Without loss of generality, we can assume that $z^{(2)}$ is the concatenation of τ and the transcript of the second commitment stage.)

4. In the second reveal stage, S_r^2 and R_r^2 receive as common input the commitment $z^{(2)}$ and a string $\sigma^{(2)} \in \{0,1\}^k$, and S_r^2 receives as private input r_S. At the end, R_r^2 accepts or rejects.

- $S = (S^1, S^2) = ((S_c^1, S_r^1), (S_c^2, S_r^2))$ and $R = (R^1, R^2) = ((R_c^1, R_r^1), (R_c^2, R_r^2))$ are computable in probabilistic polynomial time.

- We say that (S, R) is *public-coin* if it is public-coin for R.

Note that instead of providing S with decommitment values as private outputs of the commitment phases, we simply provide it with the same coin tosses throughout (so it can recompute any private state from the transcripts of the previous phases).

As for standard commitment schemes, we define the security of the sender in terms of a hiding property. Loosely speaking, the hiding property for a 2-phase commitment scheme says that *both* commitment phases are hiding. Note that since the phases are run sequentially, the hiding property for the second commitment stage is required to hold even given the receiver's view of the first stage.

Definition 2.4 (hiding)**.** 2-phase commitment scheme (S, R), with security parameter n and message length $k = k(n)$, is *statistically hiding* if for all adversarial receiver R^*,

1. The views of R^* when interacting with the sender in the first phase on any two messages are statistically indistinguishable. That is, for all $\sigma^{(1)}, \widetilde{\sigma}^{(1)} \in \{0,1\}^k$, $\mathsf{view}_{R^*}(S_c^1(\sigma^{(1)}), R^*)(1^n)$ is statistically indistinguishable to $\mathsf{view}_{R^*}(S_c^1(\widetilde{\sigma}^{(1)}), R^*)(1^n)$.

2. The views of R^* when interacting with the sender in the second phase are statistically indistinguishable no matter what the sender committed to in the first phase. That is, for all $\sigma^{(1)}, \sigma^{(2)}, \widetilde{\sigma}^{(2)} \in \{0,1\}^k$, $\mathsf{view}_{R^*}(S_c^2(\sigma^{(2)}), R^*)(\Lambda, 1^n)$ is statistically indistinguishable to $\mathsf{view}_{R^*}(S_c^2(\widetilde{\sigma}^{(2)}), R^*)(\Lambda, 1^n)$.

We stress that the second condition of the above hiding definition (Definition 2.4) requires that the view of receiver in the second phase be indistinguishable for any two messages even given the transcript of the first phase, $\Lambda = \mathsf{transcript}(S^1(\sigma^{(1)}), R^*)(1^n)$.

Loosely speaking, the binding property says that *at least* one of the two commitment phases is (computationally) binding. In other words, for every polynomial-time sender S^*, there is at most one "bad" phase $j \in \{1,2\}$ such that given a commitment $z^{(j)}$, S^* can open $z^{(j)}$ successfully both as $\sigma^{(1)}$ and $\widetilde{\sigma}^{(1)} \neq \sigma$ with nonnegligible probability. Actually, we allow this bad phase to be determined dynamically by S^*. Moreover, we require that the second phase be *statistically* binding if the sender breaks the first phase. Our construction achieves this stronger property, and using it simplifies some of our proofs.

Definition 2.5 (1-out-of-2-binding)**.** 2-phase commitment scheme (S, R), with security parameter n and message length $k = k(n)$, is *computationally* $\binom{2}{1}$*-binding* if there exist a set $\mathcal{B}$ of first phase transcripts and a negligible function ε such that:

1. For every (even unbounded) sender S^*, the first-phase transcripts in $\mathcal{B}$ make the second phase statistically binding, i.e. $\forall S^*, \forall \tau \in \mathcal{B}$, with probability at least $1 - \varepsilon(n)$ over $z^{(2)} = (S^*, R_c^2)(\tau)$, there is at most one value $\sigma^{(2)} \in \{0,1\}^k$ such that $\mathtt{output}(S^*, R_r^2)(z^{(2)}, \sigma^{(2)}) = \mathtt{accept}$.

2. $\forall$ nonuniform PPT S^*,[7] S^* succeeds in the following game with probability at most $\varepsilon(n)$ for all sufficiently large n:

   (a) S^* and R_c^1 interact and output a first-phase commitment $z^{(1)}$.

   (b) S^* outputs two full transcripts τ and $\widetilde{\tau}$ of *both* phases with the following three properties:

   - Transcripts τ and $\widetilde{\tau}$ both start with prefix $z^{(1)}$.
   - The transcript τ contains a successful opening of $z^{(1)}$ to the value $\sigma^{(1)} \in \{0,1\}^k$ using a first-phase transcript not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in τ.
   - The transcript $\widetilde{\tau}$ contains a successful opening of $z^{(1)}$ to the value $\widetilde{\sigma}^{(1)} \in \{0,1\}^k$ using a first-phase transcript not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in $\widetilde{\tau}$.

   (c) S^* succeeds if all of the above conditions hold and $\sigma^{(1)} \neq \widetilde{\sigma}^{(1)}$.

3 Our Results

Our main theorem, Theorem 1.1, is established via the following theorems.

[7] Definitions of cryptographic primitives in the literature often use the reverse order of quantifiers, asking that for every (nonuniform) PPT adversary S^*, there exists a negligible function $\varepsilon(n)$ such that the success probability of S^* is at most $\varepsilon(n)$. However, the two resulting definitions turn out to be equivalent [Bel].

Theorem 3.1. *If one-way functions exist, then on security parameter n, we can construct in time $\mathrm{poly}(n)$ a collection of public-coin 2-phase commitment schemes $\mathrm{Com}_1, \cdots, \mathrm{Com}_m$ for $m = \mathrm{poly}(n)$ such that:*

- *There exists an index $i \in [m]$ such that scheme Com_i is statistically hiding.*
- *For every index $i \in [m]$, scheme Com_i is computationally $\binom{2}{1}$-binding.*

Theorem 3.2. *Assume that on security parameter n, we can construct in time $\mathrm{poly}(n)$ a collection of public-coin 2-phase commitment schemes $\mathrm{Com}_1, \cdots, \mathrm{Com}_m$ for $m = \mathrm{poly}(n)$ such that:*

- *There exists an index $i \in [m]$ such that scheme Com_i is statistically hiding.*
- *For every index $i \in [m]$, scheme Com_i is $\binom{2}{1}$-computationally binding.*

Then, every language in **NP** *has a public-coin statistical zero-knowledge argument system.*

The proof of Theorem 3.2 is very similar to that in [NV] for $\binom{2}{1}$-statistically binding commitments, with a bit more work to handle the *computational* binding property. Thus in the rest of this abstract we describe the ideas behind the proof of Theorem 3.1. A full Full proofs for both theorems can be found in the full version of the paper [NOV].

4 Warm-up: 1-out-of-2-Binding Commitments from Regular One-Way Functions

As a warm-up to the general construction from any one-way functions, we first describe a standard commitment scheme from a regular one-way function with known preimage size (based on [HHK$^+$]), and then show how to construct a collection of statistically hiding, computationally $\binom{2}{1}$-binding commitments from regular one-way functions with unknown pre-image size.

The tools used in these commitments schemes are pairwise-independent hash functions and interactive hashing protocols, both described in the next subsections.

4.1 Hashing and Randomness Extraction

Entropy. The *entropy* of a random variable X is $H(X) = \mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}[\log(1/\Pr[X = x])]$, where here and throughout the paper all logarithms are to base 2. Intuitively, $H(X)$ measures the amount of randomness in X *on average* (in bits). The *min-entropy* of X is $H_\infty(X) = \min_x[\log(1/\Pr[X = x])]$; this is a "worst-case" measure of randomness. In general $H_\infty(X) \leq H(X)$, but if X is flat (i.e. uniform on its support), then $H(X) = H_\infty(X) = \log|\mathrm{Supp}(X)|$.

A family of hash functions $\mathcal{H}_{a,b} = \{h : \{0,1\}^a \to \{0,1\}^b\}$ is *pairwise independent* if for any two $x \neq x' \in \{0,1\}^a$ and any two $y, y' \in \{0,1\}^b$, when we randomly choose $h \leftarrow \mathcal{H}_{a,b}$, we have: $\Pr[h(x) = y \wedge h(x') = y'] = \frac{1}{2^{2b}}$. We define $\ell(a,b)$ to be the number of bits required to describe an element of the hash function family $\mathcal{H}_{a,b}$; that is, $\ell(a,b) = \max\{a,b\} + b$. We will use the following strong extractor property of $\mathcal{H}_{a,b}$.

Lemma 4.1 (Leftover Hash Lemma [BBR, ILL])**.** *Let $\mathcal{H}_{a,b}$ be a pairwise independent family of hash functions mapping $\{0,1\}^a$ to $\{0,1\}^b$. Let Z be a random variable taking values in $\{0,1\}^a$ such that $H_\infty(Z) \geq b + 2\log(1/\varepsilon)$. Then the following distribution has statistical difference at most ε from the uniform distribution on $\mathcal{H}_{a,b} \times \{0,1\}^b$: Choose $h \leftarrow \mathcal{H}_{a,b}$ and $x \leftarrow Z$ and output $(h, h(x))$.*

4.2 Interactive Hashing

Ostrovsky, Venkatesan and Yung [OVY] introduced a powerful tool known as *interactive hashing (IH)*, which is a protocol between a sender S_{IH} and receiver R_{IH}. The sender begins with a private input y, and at the end both parties outputs y_0 and y_1 such that $y \in \{y_0, y_1\}$. Informally, the IH protocol has the following properties:

1. *(Hiding)* If the sender's input y is uniformly random, then the receiver does not learn which of y_0 or y_1 equals to y.
2. *(Binding)* The sender can "control" the value of at most one of the two outputs.

Naor, Ostrovsky, Venkatesan and Yung [NOVY] showed that interactive hashing can be used to construct statistically hiding commitment schemes from one-way permutations.

We extend the notion of interactive hashing to allow multiple outputs (instead of just two output strings). Since we allow the number of outputs to be possibly superpolynomial, we succinctly describe the set of outputs as the image of a polynomial-sized circuit $C\colon \{0,1\}^k \to \{0,1\}^q$, where k and q are polynomially related to the security parameter.

For a relation W, let $W_y = \{z : W(y,z) = 1\}$ and we refer to any $z \in W_y$ as a *valid witness* for y. In the definitions below, we use general relations, and hence do not require that relation W be polynomial-time computable.

Definition 4.2. An *interactive hashing scheme with multiple outputs* is a polynomial-time protocol $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ where both parties receive common inputs $(1^q, 1^k)$, S_{IH} receives a private input $y \in \{0,1\}^q$, with the common output being a circuit $C\colon \{0,1\}^k \to \{0,1\}^q$, and the private output of S_{IH} being a string $z \in \{0,1\}^k$. We denote q to be the input length and k to be the output length. The protocol $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ has to satisfy the following security properties:

1. *(Correctness)* For all R^* and all $y \in \{0,1\}^q$, letting $C = (S_{\mathrm{IH}}(y), R^*)(1^q, 1^k)$ and $z = \mathsf{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(y), R^*)$, we have that $C(z) = y$.

2. *(Perfect hiding)* For all R^*, (V, Z) is distributed identically to (V, U_k), where $V = \mathsf{view}_{R^*}(S_{\mathrm{IH}}(U_q), R^*)$ and $Z = \mathsf{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(U_q), R^*)$.

3. *("Computational" binding)* There exists an oracle PPT algorithm A such that for every S^* and any relation W, letting circuit $C = (S^*, R_{\mathrm{IH}})(1^q, 1^k)$ and $((x_0, z_0), (x_1, z_1)) = \mathsf{output}_{S^*}(S^*, R_{\mathrm{IH}})$, if it holds that

   $$\Pr[x_0 \in W_{C(z_0)} \wedge x_1 \in W_{C(z_1)} \wedge z_0 \neq z_1] > \varepsilon,$$

   where the above probability is over the coin tosses of R_{IH} and S^*. Then we have that

   $$\Pr_{y \leftarrow \{0,1\}^q}[A^{S^*}(y, 1^q, 1^k, \varepsilon) \in W_y] > 2^{-k} \cdot (\varepsilon/q)^{O(1)}.$$

We make three remarks regarding the above definition.

1. The security requirements should hold for all, even computationally unbounded R^* (for correctness and perfect hiding) and computationally unbounded S^* (even though binding is "computational"). In addition, the relation W need not be polynomial-time computable.

2. To simplify notation, we often write $A^{S^*}(y)$, or even $A(y)$, to denote $A^{S^*}(y, 1^q, 1^k, \varepsilon)$.

3. Although the output of the honest sender S_{IH} is always a string z, the output of the cheating sender S^* is arbitrary; hence, we can assume without loss of generality that S^* breaks binding by producing two pairs of strings (x_0, z_0) and (x_1, z_1).

We think of the string $z \in \{0,1\}^k$ as a k-bit string commitment associated to one of the 2^k outputs strings, namely $y = C(z)$, and a witness $x \in W_y = W_{C(z)}$ as a decommitment to z. Intuitively, the knowledge of x gives the sender the ability to decommit to z. The "computational" binding property, read in its contrapositive, says that if it is hard to find a witness for a uniformly random string y, then it is hard for a sender to successfully decommit to two different values. Notice that this property holds even if the set of "hard" y's is not fixed in advance, but depends on the algorithm trying to find a witness for y (i.e. an element in W_y). In several places, however, we will only need the special case of a static set of y's as captured in the following lemma.

Lemma 4.3 (binding for static sets)**.** *For any protocol $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ satisfying the computational binding condition of Definition 4.2, the following holds: For all S^* and any set $T \subseteq \{0,1\}^q$, letting $C = (S^*, R_{\mathrm{IH}})(1^q, 1^k)$, we have*

$$\Pr[\exists z_0 \neq z_1 \text{ s.t. } C(z_0), C(z_1) \in T] < (\mu(T) \cdot 2^k)^{\Omega(1)} \cdot \mathrm{poly}(q),$$

where the above probability is taken over the coin tosses of S^* and R_{IH}.

Compare the bound of the above lemma to the case where adversarial sender S^* had control of only one output string. This means that the rest of the $2^k - 1$ outputs strings are distributed uniformly on $\{0,1\}^q$, and hence the bound would be $\mu(T) \cdot (2^k - 1)$. (S^* will make the string that it controls lie in T, and the probability that at least one of the rest of the $2^k - 1$ strings lie in T is at most $\mu(T) \cdot (2^k - 1)$, by a union bound argument.) The above bound is almost as good, and in particular if $\mu(T)$ is negligible and k logarithmic, both probabilities are negligible.

We extend the theorem in [NOVY] to obtain the following theorem. The protocol is obtained by simply ending the NOVY protocol $k - 1$ rounds earlier.

Theorem 4.4. *There exist interactive hashing schemes with multiple outputs satisfying Definition 4.2.*

4.3 From Regular One-Way Function with Known Preimage Size

We first informally describe a (standard) commitment scheme from a regular one-way function with known preimage size and known hardness $s(n) = n^{\omega(1)}$, based loosely on Haitner et al. [HHK$^+$] (who prove a stronger result, not needing to know the hardness).

Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a regular one-way function such that the entropy $H(f(U_n)) = t$ is known (this is equivalent to knowing the preimage size of f). In the commitment scheme, the sender S generates a random string $x \in \{0,1\}^n$ and sets $y = f(x)$. S picks a random hash function $h\colon \{0,1\}^n \to \{0,1\}^{t-\Delta}$ where $\Delta = (\log s(n))/2$. (S, R) then run the interactive hashing protocol (with $k = 1$) with S having input $(h, h(y))$. Their common output is a pair $(w_0, w_1) = (C(0), C(1))$, and the sender receives $d \in \{0,1\}$ such that $w_d = w$. To commit to the bit b, S sends $c = d \oplus b$. The commitment z is defined as (w_0, w_1, c). In the reveal phase, S sends b, d, and the string $x \in \{0,1\}^n$ used to generate y. R checks that $f(x) = y$, $c = d \oplus b$, and w_d is of the form $(h, h(y))$.

Intuitively, the commitment scheme is hiding since there are 2^t possible values of y hence $(h, h(y))$ is $(1/s(n))^{\Omega(1)}$-close to the uniform distribution by the Leftover Hash Lemma (Lemma 4.1), which implies that the commitment scheme is hiding by the hiding property of interactive hashing. As for the binding property, the one-wayness of f intuitively guarantees that the set T of y's for which a sender

S^* can compute an element of $f^{-1}(y)$ is of density at most $2^{-s(n)}$ in $\mathrm{Image}(f)$, i.e. of size at most $2^{H(f(U_n))-s(n)}$. Thus the set of pairs $(h, h(y))$ such that $y \in T$ has density at most $2^{H(f(U_n))-s(n)}/2^{t-\Delta} = s(n)^{9/10} = \mathrm{neg}(n)$. By the binding property of interactive hashing (Lemma 4.3), the probability that S^* can force both $w_0, w_1 \in T$ is negligible and the scheme is computationally binding. (The complete argument to prove the binding property is actually more subtle because the set T is not actually fixed in advance, and we need to use the computational binding property of interactive hashing given in Definition 4.2)

4.4 From Regular One-Way Function with Unknown Preimage Size

We show that if regular one-way functions with known hardness exist, then on security parameter 1^n, we can construct a collection of 2-phase commitment schemes $\mathrm{Com}_1, \cdots, \mathrm{Com}_n$ such that:

- There exists an index $i \in [n]$ such that scheme Com_i is statistically hiding.
- For every index $i \in [n]$, scheme Com_i is $\binom{2}{1}$-computationally binding.

To deal with the case where the preimage size is unknown, a first attempt would be to try all possible values of t in the protocol sketched above in Section 4.3 and obtain a collection of standard commitments. However, the above commitment scheme only seems to be computationally binding when $t \gtrsim H(f(U_n))$ (and is hiding when $t \lesssim H(f(U_n))$) not matching the guarantees of the desired collection of commitments.

We will in fact use the above protocol as the first phase. However, we also introduce a *second* phase that will be binding when $t \lesssim H(f(U_n))$ and hiding when $t \gtrsim H(f(U_n))$. This will be obtained by the sender using (a hash of) the preimage x as an input to another execution of interactive hashing. Note that given $y = f(x)$, x is distributed uniformly over a set of size $|f^{-1}(y)| = 2^{n-H(f(U_n))}$ so hiding and binding follow from the properties of interactive hashing. In fact these schemes for regular one-way functions achieve a stronger property than $\binom{2}{1}$-binding. For each value of t, either the first phase is always binding or the second phase is always binding (i.e. the sender cannot choose which binding property to break). However, we will in fact show that $\binom{2}{1}$-binding in the sense of Definition 2.5 is achieved for *any* one-way function f, regardless of whether it is regular. We use this $\binom{2}{1}$-binding commitment for each possible value of t. This ensures that all are $\binom{2}{1}$-binding and at least one of the commitments in this collection is hiding.

4.5 The Protocol

Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be any function, not necessarily regular nor one-way—as we shall later see, the regularity condition and one-way security of the function give us the hiding and binding properties, respectively. Let $\mathcal{H}_{a,b} = \{h_{a,b}\colon \{0,1\}^a \to \{0,1\}^b\}$ be a family of pairwise hash functions. The description of each element in $\mathcal{H}_{a,b}$ takes $\ell(a,b) = \max\{a,b\} + b < 2(a+b)$ bits. For $a, b < \mathrm{poly}(n)$, it is convenient to make $\ell(a,b) = q(n) - b$, for some fixed polynomial $q(n)$, so that for every $y \in \{0,1\}^a$, $|h, h(y)| = q(n)$. This can be done by padding random bits to the description of h.

In addition, it will be convenient to work with protocols where the sender has no input σ to be committed to, but rather privately receives an output $d \in \{0,1\}^k$ at the end of each phase of the commitment. If we can ensure that d is (nearly) uniform given the receiver's view, such a protocol can be tuned into a standard commitment scheme, where the sender can commit to an σ of its choice by sending $d \oplus \sigma$ at the end of the commit phase.

The following two lemmas establish the statistical hiding and computational $\binom{2}{1}$-binding properties of Protocol 4.5. The proofs of the lemmas are in the full version of this paper [NOV].

Lemma 4.6 (statistical hiding)**.** *If f is a regular function with $H(f(U_n)) \in (t_0 - 1, t_0]$, then Protocol 4.5, with setting of parameters $t = t_0$, $k \leq q(n)$, and $\Delta_1 = \Delta_2 = \omega(\log n)$, is statistically hiding in the sense of Definition 2.4.*

Lemma 4.7 (computational 1-out-of-2-binding)**.** *If f is a $s(n)$-secure one-way function (not necessarily regular), then for any value of $t \in \{1, \cdots, n\}$, Protocol 4.5, with setting of parameters $k = O(\log n)$, $\Delta_1 = \Delta_2 \leq (\log(s(n)))/4$, is computationally $\binom{2}{1}$ binding in the sense of Definition 2.5.*

5 Overview of Construction for General One-Way Functions

We now present an overview of how we generalize our construction for regular one-way functions with unknown preimage size (Protocol 4.5) to arbitrary one-way functions. As shown in Lemma 4.7, this protocol already achieves $\binom{2}{1}$-binding when f is any one-way function (for every value of t). Thus the challenge is the hiding property. (Another issue is that Protocol 4.5 requires a one-way function with known security. It turns out that our method for handling the hiding property also eliminates the need to know the security.)

As discussed in Section 4, for regular one-way functions, Protocol 4.5 has a hiding first phase when the parameter t satisfies $t \lesssim H(f(U_n))$ and has a hiding second

Protocol 4.5. 2-Phase Commitment Scheme (S, R) based on $f\colon \{0,1\}^n \to \{0,1\}^n$.

Parameters: Integers $t \in \{1, 2, \ldots, n\}$, $k \in \{1, 2, \ldots, n\}$, $\Delta_1 \in \{0, 1, \ldots, t\}$, and $\Delta_2 \in \{0, 1, \ldots, n - t\}$.

Sender's private input: String $x \in \{0,1\}^n$. (Note that this is not the value to which the sender is committing, but is rather part of its coin tosses, which will be chosen uniformly at random by S unless otherwise specified.)

First phase commit:

1. S_c^1 sets $y = f(x)$.
2. Let $\mathcal{H}_1 = \{h_1\colon \{0,1\}^n \to \{0,1\}^{t-\Delta_1}\}$ be a family of pairwise independent hash functions. S_c^1 chooses a random hash $h_1 \leftarrow \mathcal{H}_1$, and computes $v = (h_1, h_1(y)) \in \{0,1\}^q$.
3. (S_c^1, R_c^1) run Interactive Hashing Scheme $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$, with S_c^1 and R_c^1 acting as S_{IH} and R_{IH} respectively.

   Let circuit $C^{(1)}\colon \{0,1\}^k \to \{0,1\}^q$ be the common output and $d^{(1)} \in \{0,1\}^k$ be S_{IH}'s private output in $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$.

First phase sender's private output: String $d^{(1)} \in \{0,1\}^k$.

First phase reveal:

S_r^1 sends the tuple $\gamma^{(1)} = (d^{(1)}, y, h_1)$.

Receiver R_r^1 accepts if and only if $C^{(1)}(d^{(1)}) = (h_1, h_1(y))$.

Second phase commit:

Second phase common input: First-phase transcript $\tau = \mathsf{transcript}(S^1(x), R^1)$, which in particular includes the string y.

1. Let $\mathcal{H}_2 = \{h_2\colon \{0,1\}^n \to \{0,1\}^{n-t-\Delta_2}\}$ be a family of pairwise independent hash functions. S_c^2 chooses a random hash $h_2 \leftarrow \mathcal{H}_2$, and computes $w = (h_2, h_2(x)) \in \{0,1\}^q$.
2. (S_c^2, R_c^2) run Interactive Hashing Scheme $(S_{\mathrm{IH}}(w), R_{\mathrm{IH}})(1^q, 1^k)$, with S_c^2 and R_c^2 acting as S_{IH} and R_{IH} respectively.

   Let circuit $C^{(2)}\colon \{0,1\}^k \to \{0,1\}^q$ be the common output and $d^{(2)} \in \{0,1\}^k$ be S_{IH}'s private output in $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$.

Second phase sender's private output: String $d^{(2)} \in \{0,1\}^k$.

Second phase reveal:

S_r^2 sends the tuple $\gamma^{(2)} = (d^{(2)}, x, h_2)$.

Receiver R_r^2 accepts if and only if $f(x) = y$ and $C^{(2)}(d^{(2)}) = (h_2, h_2(x))$.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

IEEE
COMPUTER
SOCIETY

phase when t satisfies $t \gtrsim H(f(U_n))$. Intuitively, when f is not regular, we should replace the fixed value $H(f(U_n))$ with the 'dynamic' value $H_f(y) \stackrel{\text{def}}{=} \log(1/\Pr[f(U_n) = y])$, where $y = f(x)$ is the value chosen by the sender in the pre-processing step, because $H_f(y)$ can be viewed as measuring the amount of "entropy" in y. The "approximable preimage-size one-way functions" studied by Haitner et al. [HHK$^+$] come equipped with an algorithm that estimates $H_f(y)$, but for general one-way functions, this quantity may be infeasible to compute.

A weakly hiding scheme. One natural approach is to have the sender choose t at random and "hope" that it is close to $H_f(y)$. When we choose t too small, only the first phase will be hiding, and when we choose t too large, only the second phase will be hiding. But we have a nonnegligible probability δ (specifically, $\delta = 1/n$) that $t \approx H_f(y)$, and thus both phases will be hiding. Thus we have a $\binom{2}{1}$-binding commitment scheme satisfying a "weak hiding" property, where with probability δ, both phases are hiding, and it is always the case that at least one phase is hiding. Actually, in order to simplify our analysis, we will include t as a parameter to the protocol. Then there exists a choice of t such that the protocol is weakly hiding in the sense above, and for all choices of t the protocol is $\binom{2}{1}$-binding. At the end, we will enumerate over all values of t, resulting in a *collection* of commitment schemes as claimed in Theorem 3.1, albeit with a weak hiding property.

Unfortunately, we do not know how to directly construct zero-knowledge arguments from a weakly hiding $\binom{2}{1}$-binding commitment scheme. Thus instead, much of the effort in this paper is devoted to amplifying the weak hiding property ($\delta = 1/n$) into a strong hiding property ($\delta = 1 - \text{neg}(n)$), while maintaining the $\binom{2}{1}$-binding property.

Amplifying the hiding property. Inspired by the breakthrough results of Reingold [Rei] and Dinur [Din] on different topics, we do not amplify the hiding probability from $\delta = 1/n$ to $\delta = 1 - \text{neg}(n)$ in "one shot," but instead perform a sequence of $\log n$ iterations, each one of which increases δ by a roughly factor of 2. This results in $\delta = \Omega(1)$, and then we are able to get $\delta = 1 - \text{neg}(n)$ in just one more amplification step.

How do we double δ? A natural idea is to consider several, executions of the previous weakly hiding scheme. Specifically, if we take $m = O(1)$ executions, the probability that at least one of the executions has both phases hiding is roughly $m \cdot \delta$. Moreover, each of the remaining $m - 1$ executions have either the first phase hiding or the second phase hiding. Thus for some value of β, there are $\beta + 1$ first phases that are hiding and $m - \beta$ second phases that are hiding. It turns out that we can choose β so that this exact $(\beta + 1, m - \beta)$ breakdown given that one execution has both phases hiding occurs with probability $\Omega(1/\sqrt{m})$. Thus we are in the situation described with probability $m \cdot \delta \cdot \Omega(1/\sqrt{m}) > 2\delta$, for a large enough constant m.

Now our aim is to combine the outcomes of the weakly hiding schemes in such a way that when the above-described situation occurs, which happens with probability at least 2δ, both phases are hiding. Notice that the secret values $\sigma_1, \ldots, \sigma_m \in \{0,1\}^k$ to which the sender commits in the first commit phases have entropy (even min-entropy) at least $(\beta + 1) \cdot k$ conditioned on the receiver's view (because $(\beta+1)$ of them are hiding), and similarly the sender's secrets in the second commit phases have entropy at least $(m - \beta) \cdot k$ conditioned on the receiver's view. Let us compare this to the situation with binding. Since each execution is $\binom{2}{1}$-binding, a cheating polynomial-time sender can break the binding property for either at most β of the first phases or at most $m - \beta - 1$ of the second phases. Thus the number of possible values to which the sender can open in each case is at most $2^m \cdot 2^{k \cdot \beta}$ in the first phase or at most $2^{k \cdot (m - \beta - 1)}$, where the 2^m factor in the first bound comes from the sender's ability to choose which subset of executions to break (and it is this factor that limits us to taking m to be a constant). We can view these as strong forms of "entropy" upper bounds $m + k\beta$ and $k \cdot (m - \beta - 1)$. In at least one phase, there will be an entropy gap of at least $k - m$.

How can we exploit these entropy gaps? It turns out that interactive hashing, again, is a useful tool. Specifically, in the first phase we have the sender choose a random pairwise independent hash function h_1 mapping to approximately $(\beta + 1) \cdot k$ bits and use $(h_1, h_1(\sigma_1, \ldots, \sigma_m))$ as the input to an Interactive Hashing protocol, and analogously for the second phase. By the Leftover Hash Lemma, this pairwise independent hashing converts the min-entropy lower bound described above to an almost-uniform distribution, so the Interactive Hashing hiding property applies. As for the binding property, the bound on the number of the sender's choices gets translated to saying that the sender's input (in the first phase) comes from a set T of density $2^{-(k-m)}$, so the Interactive Hashing binding property applies. The analyses for the second phase are similar. Formalizing these ideas, we get a new $\binom{2}{1}$-binding commitment scheme in which both phases are hiding with probability at least 2δ.

When we try to iterate this amplification step $O(\log n)$ times, we run into a new difficulty. Specifically, the above sketch hides the fact that we pay entropy losses of $\omega(\log n)$ in both the hiding and binding analyses. The entropy loss of $\omega(\log n)$ in the hiding property comes from the Leftover Hash Lemma, in order to ensure that $(h_1, h_1(\sigma_1, \ldots, \sigma_m))$ has negligible statistical distance from uniform. The entropy loss of $\omega(\log n)$ in the binding property comes from needing the $\mu(T) \cdot 2^k$ factor to be negligible when applying Lemma 4.3. This forces us to go, in one step of amplification, from a commitment scheme for secrets of length k to a scheme for secrets of length $k - m - \omega(\log n)$. As in Lemma 4.7, we can take the initial secret length to be $k = \Theta(\log s(n)) = \omega(\log(n))$ if the one-way function has known security $s(n) = n^{\omega(1)}$. But to tolerate $\log n$ losses of $\omega(\log n)$, we would need $s(n) = n^{\omega(\log n)}$ (i.e., at least quasipolynomial security).

To get around this difficulty, in the amplification, we work with more relaxed, "average-case" measures of "entropy" for both the hiding and binding analyses. Specifically, for hiding, we keep track of the expected collision probability of the sender's secret, conditioned on the receiver's view. (Actually, we use the expected square root of the collision probability.) For binding, we work with the expected number of values to which the sender can open. In both cases, we only require these expectations to be within a constant factor of the ideal values (2^{-k} and 1 respectively). With these measures, it turns out that we need only lose $O(m) = O(1)$ bits in the entropy gap with each amplification step. Moreover, once we amplify δ to a constant, we can afford to take the number of executions m to equal the security parameter n and get an $\Omega(n)$-bit "entropy gap" in the final amplification step. This allows us to achieve exponentially strong statistical hiding even when we do not know the security and start with secret length of $k = O(\log n)$.

The hiding analysis of the above construction works only for certain values of t in the weakly hiding scheme, and for certain values of the β's in the amplification steps. We try out all possible values of t and β's, thus obtaining a collection $\mathrm{poly}(n)$ schemes, at least one of which is strongly hiding and all of which are $\binom{2}{1}$-binding. Notice that the number of possible choices of t and the β's are polynomial in n since $t \in \{1, 2, \ldots, n\}$, the β's in the each step except for the last is in the range $\{0, 1, \ldots, m-1\}$, for some constant m, and the last β is in the range $\{0, 1, \ldots, n\}$.

Acknowledgements. We thank Oded Goldreich, Alex Healy, Rafail Ostrovsky and Omer Reingold for helpful discussions.

References

[AH] W. Aiello and J. Håstad. Statistical Zero-Knowledge Languages Can Be Recognized in Two Rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[BLV] B. Barak, Y. Lindell, and S. Vadhan. Lower Bounds for Non-Black-Box Zero Knowledge. Technical Report TR04–083, ECCC, 2004. Extended abstract in *FOCS '04*.

[Bel] M. Bellare. A note on negligible functions. *J. Cryptology*, 15(4):271–284, 2002.

[BBR] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988. Special issue on cryptography.

[BHZ] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP Have Short Interactive Proofs? *Info. Proc. Lett.*, 25(2):127–132, 1987.

[BKK] J. Boyar, S. A. Kurtz, and M. W. Krentel. A Discrete Logarithm Implementation of Perfect Zero-Knowledge Blobs. *J. Cryptology*, 2(2):63–76, 1990.

[BCC] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BCY] G. Brassard, C. Crépeau, and M. Yung. Constant-round perfect zero-knowledge computationally convincing protocols. *Theor. Comput. Sci.*, 84(1):23–52, 1991.

[Dam] I. Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In *Proc. EUROCRYPT*, pages 203–216, 1987.

[DPP] I. B. Damgård, T. P. Pedersen, and B. Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Trans. Info. Theory*, 44(3):1143–1151, 1998.

[DH] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Trans. Info. Theory*, 22(6):644–654, 1976.

[Din] I. Dinur. The PCP Theorem via Gap Amplification. In *Proc. 38th STOC*, 2006.

[For] L. Fortnow. The Complexity of Perfect Zero-Knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989.

[Gol1] O. Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *J. Cryptology*, 6(1):21–53, 1993.

[Gol2] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[GGM] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *J. ACM*, 33(4):792–807, 1986.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE

IEEE
COMPUTER
SOCIETY

[GK] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

[GMW] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *J. ACM*, 38(1):691–729, 1991.

[GMR1] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GMR2] S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[HHK$^+$] I. Haitner, O. Horvitz, J. Katz, C.-Y. Koo, R. Morselli, and R. Shaltiel. Reducing Complexity Assumptions for Statistically-Hiding Commitment. In *Proc. EUROCRYPT*, pages 58–77, 2005.

[HILL] 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.

[ILL] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random Generation from one-way functions. In *Proc. 21st STOC*, pages 12–24, 1989.

[IL] R. Impagliazzo and M. Luby. One-way Functions are Essential for Complexity Based Cryptography. In *Proc. 30th FOCS*, pages 230–235, 1989.

[IR] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. 21st STOC*, pages 44–61, 1989.

[KSS] J. Kahn, M. Saks, and C. Smyth. A dual version of Reimer's inequality and a proof of Rudich's conjecture. In *15th Annual IEEE Conference on Computational Complexity*, pages 98–103, 2000.

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

[NOVY] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect Zero-Knowledge Arguments for NP Using Any One-Way Permutation. *J. Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO '92*.

[NY] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In *Proc. 21st STOC*, pages 33–43, 1989.

[NOV] M. Nguyen, S. J. Ong, and S. Vadhan. Statistical Zero-Knowledge Arguments for NP from Any One-Way Function. Technical Report TR06-075, ECCC, 2006.

[NV] M. Nguyen and S. Vadhan. Zero Knowledge with Efficient Provers. In *Proc. 38th STOC*, 2006.

[Ost] R. Ostrovsky. One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 133–138, 1991.

[OVY] R. Ostrovsky, R. Venkatesan, and M. Yung. Fair Games Against an All-Powerful Adversary. *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 155–169, 1993.

[OW] R. Ostrovsky and A. Wigderson. One-Way Functions are Essential for Non-Trivial Zero-Knowledge. In *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems*, pages 3–17, 1993.

[Rei] O. Reingold. Undirected ST-connectivity in log-space. In *Proc. 37th STOC*, pages 376–385, 2005.

[Rom] J. Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. In *Proc. 22nd STOC*, pages 387–394, 1990.

[Rud] S. Rudich. *Limits on the Provable Consequences of One-Way Functions*. PhD thesis, U.C. Berkeley, 1988.

[Sha] C. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[Sim] D. Simon. Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In *Proc. EUROCRYPT*, pages 334–345, 1998.

[Vad] S. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, August 1999.

Proceedings of the 47th Annual IEEE Symposium
on Foundations of Computer Science (FOCS'06)
0-7695-2720-5/06 $20.00 © 2006 IEEE