Perfect Zero-Knowledge Arguments for NP Can Be Based on General Complexity Assumptions

(EXTENDED ABSTRACT)

Moni Naor[1], Rafail Ostrovsky[2*], Ramarathnam Venkatesan[3], Moti Yung[4]

[1] IBM Research Division, Almaden Research Center, San Jose, CA
[2] International Computer Science Institute at Berkeley and University of California at Berkeley
[3] Bellcore, 445 South Street, Morristown, N.J. 07960.
[4] IBM Research, T.J. Watson Research Center, Yorktown Heights, NY 10598.

Abstract. "Zero-knowledge arguments" is a fundamental cryptographic primitive which allows one polynomial-time player to convince another polynomial-time player of the validity of an NP statement, without revealing any additional information in the information-theoretic sense. Despite their practical and theoretical importance, it was only known how to implement zero-knowledge arguments based on specific algebraic assumptions; basing them on a general complexity assumption was open since their introduction in 1986 [BCC, BC, CH]. In this paper, we finally show a general construction, which can be based on *any* one-way permutation.

We stress that our scheme is *efficient*: both players can execute only polynomial-time programs during the protocol. Moreover, the security achieved is *on-line*: in order to cheat and validate a false theorem, the prover must break a cryptographic assumption on-line *during the conversation*, while the verifier can not find (ever!) any information unconditionally (in the information theoretic sense).

* Part of this work was done while visiting Bellcore, and part at IBM T.J. Watson Research Center.

E.F. Brickell (Ed.): Advances in Cryptology - CRYPTO '92, LNCS 740, pp. 196-214, 1993.

1 Introduction

Reducing complexity assumptions for basic cryptographic primitives is a major current research program in cryptography. Characterizing the necessary and sufficient complexity conditions needed for primitives helps us develop the theoretical foundations of cryptography, and further, reducing requirements for a primitive may imply more concrete underlying functions for its practical implementations.

Here we study the problem of secure transfer of the proof of "validity of an NP assertion" in this perspective. We note that the ability to convey proofs for NP in a secure way (i.e., in *zero-knowledge* (ZK) fashion, as defined by [GMR]) has a large variety of applications in cryptography and distributed computing.

Informally, proving some fact in zero-knowledge is a way for one player (called "prover") to convince another player (called "verifier") that certain fact is true, while not revealing any additional information. In our setting, we assume that both players are polynomially bounded (thus NP proofs where the prover has a witness, are the natural setting). We must make complexity assumptions for implementing the above task since in our setting these protocols imply existence of a one-way function. The assumptions could be used in two different ways:

1. Zero-knowledge *proofs* [GMR, GMW]: The prover can not convince the verifier to accept a false theorem, even if he gets help from an infinitely powerful computation; while the verifier (or anyone overhearing the protocol), if he ever breaks the assumption (say, after 100 years), *can* extract additional information about the proof (thus, the security is only ensured computationally).
2. Zero-knowledge *arguments* [CH, BC, BCC]: The verifier can not extract additional information even if he is given infinite time (i.e., security is perfect); however, the prover (assumed to be polynomial-time) can cheat in his proof only if he manages to break the assumption *on-line during the execution of the protocol.* This is the reason to call it an "argument" rather than a "proof".

In many practical settings, ZK-arguments may be preferable to ZK- proofs: the verifier must only be sure that the prover did not break the assumption *during their interaction* (which lasted, say, ten seconds or minutes). Notice that while assuring that the assumption can *never* be broken is unreasonable, the assumption that something can not be broken *during the next ten minutes* can

be based on the current state of the art. On the other hand, the prover has absolute (i.e. information-theoretic) guarantee that no additional information is released, even if the verifier spends as much time as it desires trying (off-line) to extract it. (Thus, the notion of zero-knowledge arguments is useful if there is a need to maintain the secrecy for very long time independent of the possible future advance of cryptanalysis).

So far the complexity assumptions needed for perfect-zero-knowledge arguments were too strong — they required specific algebraic assumptions. This is in contrast with zero-knowledge interactive proofs, which can be based on any one-way function. In this work we finally dispose of specific algebraic assumptions for zero-knowledge arguments:

Main result: If one-way permutations exist, then it is possible for polynomial-time players to perform a perfect zero-knowledge arguments for all of $\mathcal{NP}$

In our proof, we construct an information-theoretically secure bit-commitment scheme, which has additional applications like information-theoretically secure coin-flipping. We can implement the scheme (with almost-perfect security) based on k-regular one-way functions. One practical implication of our result is that secure arguments can now be based on functions which are DES-like ciphers.

1.1 Background and organization

Past successes in establishing basic cryptographic primitives on general assumptions (initiated in [Y82]) have shown that various primitives, which were originally based on specific algebraic functions, can be based on the existence of general one-way functions or permutations. For example, Naor [N] showed that computationally secure bit commitments (i.e., bit commitments which *can be* broken off-line given sufficient resources) can be constructed from a pseudo-random generators (a notion originated and first implemented based on a discrete logarithm assumption in [BM]). The later, in turn (after a long sequence of papers) can now be based on any one-way function [ILL, H]. Another primitive that can now be based on any one-way function as well is digital-signature [NY, Ro]. Furthermore these primitives (and primitives derived from them, e.g. identification) were shown to imply a one-way function (thus they are equivalent) [IL]. On the other hand, basing the primitive of oblivious transfer on a general one-way permutation which is not a trapdoor[5] was shown to be "a seemingly

[5] a trapdoor implies that there is an information which enables easy inversion

hard task" [IR] – when based on black box reductions, it will separate P and NP (on the positive side, a trapdoor permutation is sufficient).

Concerning secure proofs, Goldreich, Micali and Wigderson showed that zero-knowledge *proofs* for $\mathcal{NP}$ can be done and require secure encryption functions (the results of [N, ILL, H] give such functions under any one-way function); this applies to general $\mathcal{IP}$ proofs as well [IY]. Further, zero-knowledge proofs and zero-knowledge arguments for non-trivial languages as well as non-interactive zero-knowledge proofs of [BFM, BDMP] imply the existence of one-way functions [OW].

In contrast to computational zero-knowledge *proofs*, the primitive of perfect zero-knowledge *arguments* for NP was much inferior in this respect: their constructions were known only under specific algebraic assumptions [BCC, BKK, IY, BY, IN]. Our result gives the first general reduction: zero-knowledge NP-arguments can be constructed given any one-way permutation.

Our construction has two stages. First, we show how to design an information-theoretically secure bit commitment between two polynomial-time parties based on any one-way permutation (we employ a technique that can be called "interactive-hashing" introduced initially in a different model involving an all-powerful party [OVY1]). Moreover, we do it in such a way that the conversations in the commitment protocol are *simulatable* (i.e. by an expected polynomial time algorithm). Then, we apply the reduction of "perfectly-secure simulatable bit commitment" to "perfect ZK-argument". (A general scheme connecting various commitments to various ZK-systems was given in e.g. [IY] and can be used).

We note that this work differs from [OVY1] in that there the sender must be able to invert one-way functions, whereas here the sender is efficient (this is the traditional cryptographic model). In [OVY1] we deal with oblivious transfer and any technique succeeding in allowing a weak sender there, would be quite significant since it would implement oblivious transfer between polynomial time parties using one-way permutations (see [IR]).

1.2 Relation to recent work on bit-commitment

Recently, models in which parties may have power beyond polynomial-time were investigated; it is worth while pointing out the differences between the current work and the recent one. By "From Strong to Weak BC", we denote Bit-commitments (BC) protocols, in which even an infinitely-powerful "Commiter"

can not cheat, (i.e. change the value of the committed bit) except with negligible probability, but the polynomial-time "Receiver" can "see" the commitment, if he breaks the assumption. The result of [N] imply that under any one-way function, there is a (Strong-to-Weak) BC from a polynomial-time Commiter to a polynomial-time Receiver (that is, it is an efficient protocol and the underlying assumption in this case is optimal [IL]).

The work in [OVY2] investigated commitments between a strong and a polynomial-time players where the strong player actually uses its non-polynomial-time power. Thus, the main issue in that paper is how cryptographic assumptions changes and can be relaxed when the power of players differs (rather than being polynomial-time for both players, as needed in practical applications). It is shown that unless Distributional-NP=RP there is a (Strong-to-Weak) BC from a Commiter with an (NP union co-NP) power to a polynomial-time Receiver. Similarly, unless Distributional-PSPACE=RP, there is a (Strong-to-Weak) BC from a (PSPACE) Commiter to a polynomial-time Receiver. Distributional-NP is defined by Levin in the theory of average-case NP, whereas Distributional-PSPACE is a complete (in Levin's sense) problem for PSPACE under a uniform distribution. Thus, when allowing the commiter to use non-polynomial power this theoretical result relaxes the assumptions in [N].

By "from Weak to Strong BC" we denote BC in which even an infinitely-powerful "receiver" can not "see" the commitment, but the polynomial-time commiter can not change the value of the commitment if a complexity assumption holds. In [OVY2] it is also shown, based on an oblivious transfer protocols among unequal-power players introduced in [OVY1] (where interactive hashing was presented), that given any one-way function, there is a (Weak-to-Strong) BC from a polynomial-time Commiter to a (PSPACE) Receiver (and if the receiver is NP, the same holds under a one-way permutation).

The main results in [OVY1] yield oblivious transfer under one-way function when players have unequal power. The cryptographic application of [OVY1] (when both parties are polynomial time), is basing two-party secure computation with one party having information theoretic security under general *trapdoor* permutation assumption (whereas previously known under specific algebraic trapdoor functions). This is done by applying the results for one-way permutation but by adding a trapdoor property to be useful in cryptographic scenarios (so that computations are in polynomial-time).

In the current paper, we assume polynomial-time parties and do not use non-polynomial-time computations. We stress again that this is the model for cryptographic applications. Further, we make no use of trapdoor properties, as

BC's and secure interactive proofs do not need decryptions, but rather displaying of pre-images (for decommitals). Our result here for BC can be stated as: given any one-way permutation, there is an efficient (Weak-to-Strong) BC protocol from a polynomial-time Commiter to a polynomial-time Receiver (which may be stronger); the BC is simulatable and is a commitment of knowledge.

1.3 Organization of the paper

In section 2, we give the model, the formal definitions of the problem, and the assumptions. (Specifically, we present the model of interactive machines, the definitions of perfect zero-knowledge arguments, the notion of commitment, and the definition of one-way functions and permutations). In Section 3, we present the new method for basing a perfectly-secure bit commitment on a one-way permutation, and discuss its reduction to zero-knowledge arguments. In section 4 we present additional applications of our methods.

2 Model and Definitions

Let *Alice* (the prover) and *Bob* (the verifier) be interacting Turing machine [GMR, B] which share an access to a security parameter n, and a common communication tapes. Each has a private input and output tapes and a private random tape. When *Alice* and *Bob*'s programs are both polynomial time, we say that the protocol is "efficient" (we will assume this throughout), *Alice* usually has a private tape in which a "witness" to the correctness of the common input is written. We may consider *Bob* to be infinitely-powerful when he wishes to extract information from a protocol conversation, although he needs only poly time computations to execute the protocol. Both parties share an input tape of size k and and two "communication tapes": tapes for *Alice* to write in and *Bob* to read and vice versa. *Bob* has a private history tape h.

2.1 Perfect Zero-Knowledge Arguments

An NP-proof protocol with polynomial-time prover is a protocol between two polynomial time parties: a prover *Alice* and a verifier *Bob*. The parties take turns being "active", that is, reading the tapes and performing the computation, outputting a "message" on the corresponding communication tape. Both parties are probabilistic machines, (i.e., they have a read-only infinite tape of truly random bits which is private and read left-to-right). *Alice* also has a private

input with a witness to the input. (Without lose of generality, we can assume that the input is a legal satisfiability (SAT) statement, since otherwise any NP statement can be translated first to SAT, and *Alice* can translate the witness to a witness to the SAT-statement). At the end of the protocol *Bob* moves to one of two states: ACCEPT or REJECT.

Definition 1 *An NP-proof protocol with polynomial-time prover is called an* **argument** *if:*

1. *There exists a polynomial-time program (in the statement size which is a security parameter) for Alice such that given any statement in NP, Alice can always convince polynomial-time Bob (that is make Bob move to ACCEPT at the end of the interaction).*
2. *No polynomial-time Alice* interacting with Bob can convince Bob to ACCEPT, when the input is not true, except with negligible small probability (that is for a polynomial p for large enough input the error becomes smaller than* $1/p(n)$.

For an input I and history h let $CONV_{Bob^*}(I,h)$ be the random variable (depending on the parties' random tapes), which Bob^* produces throughout an interaction with *Alice*.

We note that similarly an argument can be prove "a possession of knowledge" in the sense that one formally shows that a machine employing the prover can extract a witness to the claimed NP statement [FFS, TW, BG]. (In the next version we describe this as well).

We say that two distributions μ_1 and μ_2 on $\{0,1\}^n$ are almost identical if for all polynomials $p(n)$, large enough n and for all $A \subset \{0,1\}^n$, $|\mu_1(A) - \mu_2(A)| < 1/p(n)$.

Definition 2 *An argument is* **perfectly zero-knowledge** *if: for all verifier* Bob^**, there is a simulator which is a probabilistic expected polynomial-time machine* M_{Bob^*}*, such that for any input* I*, it produces a random variable* $SIM_{Bob^*}(I,h)$ *so that the distribution of* $SIM_{Bob^*}(I,h)$ *is identical to that of* $CONV_{Bob^*}(I,h)$.

2.2 Commitment

Definition 3 *A* **bit commitment** *protocol consists of two stages:*

- *The* commit *stage: Alice has a bit b on her input tape, to which she wishes to commit to Bob. She and Bob exchange messages. At the end of the stage Bob has some information that represents b written on its output tape.*

– *The* reveal (opening) stage*: Alice and Bob exchange messages (where their output tapes from the commit stage are serving as input tapes for this stage). At the end of the exchange Bob writes on its output tape b.*

Definition 4 *To be* **perfectly-secure commitment,** *the protocol must obey the following: for* all *Turing machines Bob, for all probabilistic polynomial time Alice, for all polynomials p and for large enough security parameter n*

1. *(Security property:) After the commit stage, when Alice follows the protocol Bob cannot guess b with probability greater than* $\frac{1}{2} + \frac{1}{p(n)}$ *(even if Bob is given unbounded computational resources).*
2. *(Binding property:) After the commit stage in which Bob follows the protocol, with probability at least* $1 - \frac{1}{p(n)}$ *the polynomial-time Alice can reveal only one possible value.*

Note that the security property does not rely on *Bob* being polynomial time. In addition, if *Bob*'s algorithm can be performed in polynomial-time, we say that the bit commitment is "efficient"– we concentrate on this case.

We say that a commitment scheme is *polynomial-time simulatable* (with respect to the receiver) if given a polynomial-time receiver Bob^*, its history of conversations is a probability space simulatable by having Bob^* taking part in a computation with an expected polynomial time machine S (as in the definition of zero-knowledge).

We call a commitment a *commitment of knowledge* if there is a polynomial-time machine X (extractor) interacting with the sender performing the commit stage, such that the probability that X outputs a bit b is close to the probability that the reveal stage outputs same bit b (assuming reveal ended successfully). (A formal definition, is postponed to the full version).

In defining the properties that a bit commitment protocol must obey, we have assumed a scenario where *Bob* cannot guess b with probability greater than $\frac{1}{2}$ prior to the execution of the commit protocol. In the more general case, *Bob* has some auxiliary input that might allow him to guess b with probability $q > \frac{1}{2}$. The definition for this case is that as a result the commit stage the advantage that *Bob* gains in guessing b is less than $\frac{1}{p(n)}$. All the results of this paper hold for this more general case as well.

2.3 One-way functions and permutations

We define the underlying cryptographic operations we assume.

Let f be a length preserving function $f: \{0,1\}^* \to \{0,1\}^*$ computable in polynomial time.

Definition 5 [One-way function.] *f is one-way if for every probabilistic polynomial time algorithm A, for all polynomials p and all sufficiently large n,*

$$Pr[f(x) = f(A(f(x))) \mid x \in_R \{0,1\}^n] < 1/p(n).$$

The above definition is of a *strong one-way function*. Its existence is equivalent to the existence of the weaker *somewhat one-way function* using Yao's amplification technique [Y82] or the more efficient method of [GILVZ] (which is applicable only to permutations or regular functions). (A somewhat one-way function has the same definition as above, but the hardness of inversion is smaller, i.e. its probability is inverse polynomially away from 1.)

If in addition f is 1-1 then we say the f is a **One-Way Permutation**. For the construction outlined in Section 3 we require a one-way permutation f. (We note that we can also employ k-regular one-way functions in our protocol, since they can be converted into an "almost a permutation" [GKL]).

3 Perfectly-Secure Simulatable Bit Commitment

We present a perfectly-secure scheme and its proof of security. The polynomial commiter generates a bit encryption which comes from two possible distributions. The commiter will be able to open the encryption only as a member of one distribution (even though the distribution are identical).

3.1 The Scheme based on any one-way permutation

Let f be a strong one-way permutation f on $\{0,1\}^n$. Let S denote the sender *Alice* (as defined in 2.1) and R the receiver *Bob* (as defined). In the beginning of the protocol, S is given a secret input bit b. $B(x,y)$ denotes the dot-product mod 2 of x and y.

Commit Stage.

Commit to a bit b.

1. The sender S selects $x \in_R \{0,1\}^n$ at random and computes $y \leftarrow f(x)$. S keeps both x and y secret from R.

2. The receiver R selects $h_1, h_2, \ldots h_{n-1} \in \{0,1\}^n$ such that each h_i is a random vector over $GF[2]$ of the form $0^{i-1}1\{0,1\}^{n-i}$ (i.e. $i-1$ 0's followed by a 1 followed by an arbitrary choice for the last $n-i$ positions). Note that $h_1, h_2, \ldots h_{n-1}$ are linearly independent over $GF[2]$
3. For j from 1 to $n-1$
   - R sends h_j to S.
   - S sends $c_j \leftarrow B(h_j, y)$ to R.
4. At this point there are exactly two vectors $y_0, y_1 \in \{0,1\}^n$ such that for $i \in \{0,1\}$, $c_j = B(y_i, h_j)$ for all $1 \leq j \leq n-1$. y_0 is defined to be the lexicographically smaller of the two vectors. Both S and R compute y_0 and y_1. Let
$$c = \begin{cases} 0 \text{ if } y = y_b \\ 1 \text{ if } y = y_{1-b} \end{cases}$$
5. S computes c and sends it to R.

Reveal Stage.

1. S sends b and x to R.
2. R verifies that $y = f(x)$ obeys $c_j = B(h_j, y)$ for all $1 \leq j \leq n-1$ and verifies that if $c = 0$, then $y = y_b$ and if $c = 1$, then $y = y_{1-b}$.

end-commit-protocol

It is clear that the protocol described above can be executed in polynomial time by both parties. In the next subsection we will see that it is indeed a perfectly secure bit commitment protocol.

3.2 Proof of security

Theorem 1. *If f is a one-way permutations exist, then the scheme presented in Section 3.1 is a perfectly-secure computationally-binding bit commitment scheme.*

Theorem 1 follows from the two theorems below, the security theorem and the binding theorem, respectively.

Theorem 2. *For any receiver R', after the* commit *stage the bit b is hidden information-theoretically.*

Proof : We can prove inductively on j, that for any choice of $h_1, h_2, \ldots h_j$ the conditional distribution of y given $h_1, h_2, \ldots h_j$ $c_1, c_2, \ldots c_j$ is uniform in the subspace defined by $h_1, h_2, \ldots h_j$ and $c_1, c_2, \ldots c_j$. Thus, at step 4 the probability that $y = y_0$ is exactly $\frac{1}{2}$. Therefore giving away c yields nothing about b. □

Theorem 3. *Assume there exists a probabilistic polynomial time $S'(n)$ that following the commit stage can reveal to a honest receiver two different values for b with non-negligible probability (over its coin-flips) $\varepsilon = \varepsilon(n)$. Then there exists a probabilistic polynomial time algorithm A that inverts f on non-negligible fraction of the y's in $\{0,1\}^n$.*

Proof : Using such an S' we now construct the algorithm A to invert f. A has a fixed polynomial time bound and it aborts if its runtime exceeds the bound. By assumption, there exists a set Ω of $\varepsilon(n)$ fraction of strings such that if the tape of S' is initialized with $\omega \in \Omega$, S' succeeds in revealing two different values for b after the commit stage of $n-1$ rounds. We may fix such an ω and view S' as deterministic. This is true, since one can repeatedly run A with the random tape of S' initialized with $\omega_i, i := 1, \ldots, m = 1/\varepsilon^2$ and with probability $1 - e^{-\sqrt{m}}$ some $\omega_i \in \Omega$. We treat S' as a deterministic algorithm from now on.

The responses c_i of S' to the queries h_i sent by R define a rooted tree T whose edges are labeled in $\{0,1\}$. A path from the root to a leaf is defined by an assignment to $h_1, h_2, \ldots h_{n-1}$ and it is labeled with $c_1, c_2, \ldots c_{n-1}$. A node U at level i corresponds to a state of S' after $i-1$ stages. It defined by $h_1, \ldots, h_{i-1}$ and $c_1, \ldots, c_{i-1}$. The outgoing edges of U correspond to R's 2^{n-i} possible queries. These edges are labeled with the responses of S'. Note that since S' may be cheating, his answers need not be consistent and that on the same query S' may give different answers depending on the previous queries.

For a leaf u, let $\{y_0(u), y_1(u)\}$ be the set consistent with S's answers; we say u is *good* if given that R's queries define u, then S' succeeds in opening the bit committed in two different ways: i.e. S' inverts on both $y_0(u)$ and $y_1(u)$.

Description of A: A gets as an input a random image y in $\{0,1\}^n$ and it attempts to invert y. In order to compute $f^{-1}(y)$, A tries to find a good leaf u such that $y \in \{y_0(u), y_1(u)\}$. Starting at the root, A develops node by node a path consistent with y. Fix j to be $n - 8(\log n/\epsilon + 1)$. For j rounds A does as follows: for $1 \leq i < j$ at the i round the path so far is defined by $h_1, h_2, \ldots h_{i-1}$ and the labels are $c_1, c_2, \ldots c_{i-1}$ such that $c_i = B(h_i, y)$. Now, a random h of the $0^{i-1}1\{0,1\}^{n-i}$ is chosen (note that h is linearly independent from $h_k, k < i$ is chosen. If the edge h is labeled with $B(h,y)$, then $h_i \leftarrow h$ and the path is expanded by the new node. Otherwise, S' is reset to the state before its reply, and a new candidate for h_i is chosen. This is repeated until either a success or until there are no more candidates left, in which case A aborts. If A reaches the jth level, it guesses the remaining $n-j$ queries $h_j, h_{j+1}, \ldots h_{n-1}$ and checks whether the path to the leaf is labeled consistently with $B(y, h_i)$. If it is and the leaf reached is good, then A has succeeded in inverting y.

The rest of this proof is devoted for showing that $\mathcal{A}$ as defined above has probability at least $\varepsilon^{10}/8e^3n^8$ for inverting y. Note that $\mathcal{A}$ as described above does not necessarily halt after a polynomial number of steps. However, as we shall see at the end of the proof, we can limit the total number of unsuccessful attempts at finding a consistent h to $8n$ without decreasing significantly the probaiblity that $\mathcal{A}$ succeeds in inverting y.

Before we continue we introduce some notation. Since we are dealing with several types of vectors of length n over $GF[2]$ we will distinguish them by calling those vectors that are sent by R as *queries* and those vectors which may be the image that y attempts to invert as *images*. Let U be a node at the ith of the tree defined by $h_1, h_2, \ldots h_{i-1}$ and $c_1, c_2, \ldots c_{i-1}$. We say that $y \in \{0,1\}^n$ is an image in U if $B(h_k, y) = c_k$ for all $1 \leq k < i$. We denote the set of images of U by $\mathcal{I}(U)$. We know that $|\mathcal{I}(U)| = 2^{n-i+1}$. We say that $h \in \{0,1\}^n$ is a query of U if it is of the form $0^{i-1}1\{0,1\}^{n-i}$).

Let $A(U,y) = |\{h | h$ is a query of U and $B(h,y)$ agrees with the label h of $U\}|$

An image y is *balanced* in U_i, a node of the ith level if

$$1 - \frac{1}{n} \leq \frac{A(U_i, y)}{2^{n-i-1}} \leq 1 + \frac{1}{n}$$

An image y is *fully balanced* in U, a node of the jth level, if it is balanced in all the ancestors of U. Define $\mathcal{F}(U)$ as the set of all $y \in \mathcal{I}(U)$ **and** are fully balanced in U. For a set of queries H at a node U and an image y of U the *discrepancy* of y at H is the absolute difference between $|H|/2$ and the number of queries in H that agree with y. Finally, recall that $j = n - 8(\log n/\varepsilon + 1)$.

Lemma 4. *For any node U of level j at least $2^{n-j}(1-\beta)$ for $\beta = 2^{-3/4(n-j)}$ of the images of U have the property that $2^{n-j} - 2^{7/8(n-j)} \leq A(U,y) \leq 2^{n-j} + 2^{7/8(n-j)}$*

Proof : First note that any pair of queries h', h'' of U has the property that h'' is linearly independent of $h', h_1, h_2, \ldots h_{j-1}$. Now suppose that an image y of U is chosen at random and consider the indicator a_h which is 1 whenever $B(h,y)$ is equal to U's response on h. For any h we have that $Prob[a_h = 1] = 1/2$ and for every pair h', h'' the events $a_{h'}$ and $a_{h''}$ are pairwise independent. We are essentially interested in

$$Prob\left(\left| \sum_{h \text{ query of } U} a_h - E[\sum_{h \text{ query of } U} a_h] \right| \geq 2^{7/8(n-j)} \right) \qquad (1)$$

By Chebyschev's inequality

$$Prob\left(\left|\sum_{h \text{ query of U}} a_h - E[\sum_{h \text{ query of U}} a_h]\right| \geq \lambda\sqrt{VAR[\sum a_h]}\right) \leq \frac{1}{\lambda^2}$$

$Var[\sum_h a_h]$ is 2^{n-j} and hence (1) is at most $2^{-3/4(n-j)}$.

Lemma 5. *For any node U of level j and random image y of U the probability that y is fully balanced in U is at least $1-\gamma$ for $\gamma = n2^{-5/8(n-j)}$*

Proof: Let $U_1, U_2, \ldots U_j = U$ be the nodes on the path to U. For any $1 \leq i \leq j$ we can partition the 2^{n-i} queries of U_i into 2^{j-i} subsets $H_1, H_2, \ldots H_{2^{j-i}}$ of size 2^{n-j} each such that for any $1 \leq \ell \leq 2^{j-i}$ and $h', h'' \in H_\ell$ we have that h' is linearly independent of $h_{i+1}, \ldots h_j, h''$. Therefore, similar to Lemma 4, we have that $Prob[|\sum_{h\in H_\ell} - E[\sum_{h\in H_\ell} a_h]| > 2^{7/8(n-j)}] \leq 2^{-3/4(n-j)}$. Therefore by Markov's inequality the probability that more than $2^{-1/8(n-j)}$ fraction of the H_ℓ's have a discrepancy larger than $2^{7/8(n-j)}$ is at most $2^{-5/8(n-j)}$. Therefore with probability at least $1 - 2^{-5/8(n-j)}$ the total discrepancy at node U_i is at most

$$2^{-1/8(n-j)}2^{n-j}2^{j-i} + (1 - 2^{-1/8(n-j)})2^{7/8(n-j)}2^{j-i} \leq 2 \cdot 2^{7/8n+1/8j-i} \qquad (2)$$

and hence with the probability at least $1 - 2^{-5/8(n-j)}$ we have

$$1 - \frac{1}{n} \leq 1 - 2^{-1/8(n-j)+1} \leq \frac{A(U_i, y)}{2^{n-i-1}} \leq 1 + 2^{-1/8(n-j)+1} \leq 1 + \frac{1}{n}$$

The probability that y is balanced in all the levels is therefore at least $1 - n2^{-5/8(n-j)} = 1\gamma$.

Lemma 6. *The probability that a node U of the jth level is reached by an execution of A is at least $\frac{1-\gamma}{e}$ of the probability that it is reached by an execution of S'*

Proof: Let $U_1, U_2, \ldots U_j = U$ be the nodes on the path to U's from the root. For any node U_i the probability that U_i is reached in S' is $\prod_{i=1}^{j-1} \frac{1}{2^{n-i}}$. On the other hand

$$Prob[U \text{ is reached by } \mathcal{A}] = \sum_{y \in \mathcal{I}(U)} Prob[y \text{ is chosen and } U \text{ is reached}] \geq$$

$$\sum_{y\in\mathcal{F}(U)} Prob[y \text{ is chosen and } U \text{ is reached}] = \sum_{y\in\mathcal{F}(U)} 1/2^n \prod_{i=1}^{j-1} \frac{1}{A(U_i,y)} \geq$$

$$\sum_{y\in\mathcal{F}(y,U)} 1/2^n \prod_{i=1}^{j-1} \frac{1}{(1+1/n)2^{n-i-1}} \geq \sum_{y\in\mathcal{F}(U)} 1/2^n \prod_{i=1}^{j-1} \frac{1}{(1+1/n)2^{n-i-1}} \geq$$

$$\frac{2^{n-j+1}(1-\gamma)}{2^n} \cdot \frac{1}{(1+1/n)^n} \prod_{i=1}^{j-1} \frac{1}{2^{n-i-1}} \geq \frac{(1-\gamma)}{e} \prod_{i=1}^{j-1} \frac{1}{2^{n-i}}$$

Lemma 7. *The probability that the image $\mathcal{A}$ is trying to invert is fully balanced at the jth level is at least $\frac{(1-\gamma)^2}{e}$*

Proof : For every node of the jth level and every fully balanced image y of U we have that $Prob[y \text{ is chosen and } U \text{ is reached}] \geq \frac{(1-\gamma)}{2^n e} \prod_{i=1}^{j-1} \frac{1}{2^{n-i-1}}$. Hence,

$$Prob[U \text{ is reached with a fully balanced } y] \geq$$

$$2^{n-j}(1-\gamma) \cdot \frac{1-\gamma}{e2^n} \prod_{i=1}^{j-1} \frac{1}{2^{n-i-1}} = \frac{(1-\gamma)^2}{e} \prod_{i=1}^{j-1} \frac{1}{2^{n-i}}$$

The number of nodes at the jth level is $\prod_{i=1}^{j-1} 2^{n-i}$ and therefore the probability that the image chosen is fully balanced at the jth level is at least $\frac{(1-\gamma)^2}{e}$.

Call a node *good* if at least ε of the leaves at the subtree rooted at U have the property that S' succeeds in cheating, i.e., inverting both images. By assumption, the fraction of good nodes U is at least ε. Hence, by Lemma 6 the probability that $\mathcal{A}$ reaches a good U at level j is at least $\frac{1-\gamma}{e}\epsilon$.

Lemma 8. *In any good node U of level j the fraction of the good leaves that have at least one image that is in $\mathcal{F}(U)$ is at least $\varepsilon/2$.*

Proof : Any pair of images $y_1 \neq y_2$ in $\mathcal{I}(U)$ can be together in at most $1/2^{n-j}$ of the leaves: in any node U'' along the way from U to the leaves and for random query h of U' we have $Prob[B(h,y_1) = B(h,y_2)] = 1/2$. Since there are at most $\gamma 2^{n-j+1}$ images that are not fully balanced in U, then at most

$$\binom{\gamma 2^{n-j+1}}{2} / 2^{n-j-1} \leq 2\gamma^2 2^{n-j} \leq n^2 2^{-1/4(n-j)+1} = n^2 2^{-2(\log n/\varepsilon+1)+1} \leq \frac{\varepsilon^2}{2}$$

of the leaves have both of their images from the unbalanced. Therefore at least $\varepsilon - \frac{\varepsilon^2}{2} \geq \varepsilon/2$ of the leaves are both good and have at least one image which is fully balanced at U.

Lemma 9. *For any good node U of level j and $z \in \mathcal{F}(U)$, given that U was reached with a fully balanced y, the probability that $y = z$ is at least $\frac{1}{e^2 2^{n-j+1}}$*

Proof : We would like to bound from below

$$\frac{Prob[z \text{ is chosen and } U \text{ is reached}]}{Prob[U \text{ is reached and the image is fully balanced}]} \tag{3}$$

We know that $Prob[U \text{ is reached and the image is fully balanced}] =$

$$\sum_{z \in \mathcal{F}(U)} Prob[z \text{ is chosen and } U \text{ is reached}] = \sum_{y \in \mathcal{F}(U)} 1/2^n \prod_{i=1}^{j-1} \frac{1}{A(U_i, y)} \leq$$

$$\sum_{y \in \mathcal{F}(U)} 1/2^n \prod_{i=1}^{j-1} \frac{1}{(1-1/n)2^{n-i-1}} \leq \sum_{y \in \mathcal{F}(U)} e/2^n \prod_{i=1}^{j-1} \frac{1}{2^{n-i-1}} \leq$$

$$\frac{2^{n-j}}{2^n} \cdot e \prod_{i=1}^{j-1} \frac{1}{2^{n-i-1}} \leq e \prod_{i=1}^{j-1} \frac{1}{2^{n-i}}$$

As can be seen from the proof of Lemma 6 for any $z \in \mathcal{F}(U)$ we have that

$$Prob[z \text{ is chosen and } U \text{ is reached}] \geq \frac{1}{e2^{n-j+1}} \prod_{i=1}^{j-1} \frac{1}{2^{n-i}}$$

Therefore (3) is at least $\frac{1}{e^2 2^{n-j+1}}$

Lemma 10. *The probability that $\mathcal{A}$ is successful is at least $\frac{\varepsilon^{10}}{4e^3 n^8}$*

Proof : Suppose that (a) $\mathcal{A}$ reaches a good node U at level j and the y is fully balanced and (b) that $h_j, h_{j+1}, \ldots h_{n-1}$ define a path to a good leaf that has at least one image in $\mathcal{F}(U)$. Call this image z. Then by Lemma 9 we know that the probability that $y = z$ is at least $\frac{1}{e^2 2^{n-j}}$. The probability that (a) occurs is at least $\varepsilon \frac{(1-\gamma)^2}{e}$ by Lemma 7 and that (b) occurs given (a) is at least $\varepsilon/2$ by Lemma 8. Therefore the probability of success is at least $\varepsilon^2 \frac{(1-\gamma)^2}{2^{n-j} e^3} \geq \varepsilon^{10}/4e^3 n^8$

Note that we have only considered $\mathcal{A}$ successes when y was fully balanced at level j. However, given that y is fully balanced at level j, the probability that $\mathcal{A}$ had many unsuccessful candidates until he reached the jth level is small: we know that y is balanced at U_i for all $1 \leq i < j$ and therefore $A(U, y)/2^{n-i} > 1/4$. Therefore the probability that $\mathcal{A}$ had to try more than $8n$ candidate for the h_i's until reaching level j is exponentially small in n and we have that even if we bound the run time of $\mathcal{A}$ by $8n^2$ the probability of success is still at least $\varepsilon^{10}/8e^3 n^8$. If ε is non negligible, then this is non negligible as well. This concludes the Proof of Theorem 3.

For our applications we need a simulatable bit commitment and commitment of knowledge (to be defined in the full version along the lines of [BG]).

Theorem 11. There is a perfectly-secure commitment scheme which is simulatable, and is commitment of knowledge.

Proof sketch: All actions of S are in polynomial time, so simulatability (generating the same distribution in polynomial time) is given.

To achieve simulatable commitment of knowledge, one has to modify the basic protocol described above as follows. The protocol's steps 1,2, and 3 will be first performed twice. At this point R asks S to open the chosen x which is the pre-image of y of one of the instances and continue the protocol with the other instance. Obviously, the security and binding properties are maintained.

To get a commitment of knowledge, we have an extraction algorithm X which plays the steps 1,2, and 3 twice. Then, it decides on which instance to continue, it asks to open it and gets y, then the simulation is backtracked and the other instance is asked to be opened, and the actual commitment is done using the (by now known) y in step 4 and 5 (given the input bit b to the machine X). The probability that the commitment will be different is negligible assuming the hardness assumption as was shown above. □

Next, we can state the following known "reduction theorems" present in the works on computational (perfect) zero-knowledge proofs (arguments) [GMW, BCC, IY].

Theorem 12. If there is a (perfectly-secure commitment) [commitment] scheme which is simulatable by an expected probabilistic polynomial-time machine ("interacting" with the receiver) and the receiver is polynomial-time, then there is a (perfect zero-knowledge argument) [computationally zero-knowledge proof] for any statement in NP.

The perfectly-secure simulatable bit-commitment protocol can be used in the general scheme above. In addition, the general proof system scheme can also be shown to give a "proof of possession of a witness" (i.e., proof of knowledge) as was formalized [FFS, TW, BG]. Thus, combining the above, gives our main result:

Theorem 13. If any one-way permutation exists, then there exist perfect zero-knowledge arguments for proving language-membership as well as for proving knowledge-of-witness.

4 Discussion

There are various other applications to information-theoretically secure bit commitment. For example, another application of the bit commitment above is a "coin-flipping protocol" (introduced by Blum [B]), with perfect security, and assuming only a one-way permutations.

For practical purposes consider the data encryption standard (DES) [Kon]. Given a k-regular [GKL] one-way function (i.e. the number of pre-images of a point is $\leq k$ and is k on a significant fraction), one can transform it into a one-way function which is 1-1 almost everywhere [GILVZ]. We apply this to the function $DES(k,m) = y$ (k = key, m= message) where (actual used parameters are) $k \in \{0,1\}^{56}$, $m, y \in \{0,1\}^{64}$. Assuming that DES is not breakable on-line (say in 10 seconds), then it is a good candidate for our scheme. We explore this further in the full version of the paper. The security of the commitment is not perfect but rather almost-perfect (guessing the commitment is not exactly $1/2$, but it is close to $1/2$). We note that DES is available in many machines and usually on an optimized hardware circuit.

It is an interesting question whether a general one-way function with no additional property suffices for zero-knowledge arguments. Reducing the rounds (by more than the achievable logarithmic factor) is interesting as well.

Acknowledgments

We would like to thank Dalit Naor for helpful advice.

References

[BDMP] Blum M., A. DeSantis, S. Micali and G. Persiano, "Non-Interactive Zero-Knowledge" *SIAM J.Comp. 91*

[BM] Blum M., and S. Micali "How to Generate Cryptographically Strong Sequences Of Pseudo-Random Bits" *SIAM J. on Computing,* Vol 13, 1984, pp. 850-864, FOCS 82.

[BFM] Blum M., P. Feldman, and S. Micali "Non-Interactive Zero-Knowledge Proof Systems and Applications" *STOC 88.*

[BMO] Bellare, M., S. Micali and R. Ostrovsky, "The (True) Complexity of Statistical Zero Knowledge" STOC 90.

[B] Blum, M., "Coin Flipping over the Telephone," IEEE COMPCON 1982, pp. 133-137.

[BKK] J. Boyar, S. Kurtz, and M. Krental *A Discrete Logarithm Implementation of Perfect Zero-Knowledge Blobs*, Journal of Cryptology, V. 2 N. 2, 1990, pp. 63-76, Springer International.

[BG] Bellare M., and O. Goldreich, "On Defining Proof of Knowledge," *CRYPTO* 92 (this proceedings).

[BC] G. Brassard, C. Crépeau, *"Non-Transitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond"*, FOCS 86 pp. 188-195.

[BCC] G. Brassard, D. Chaum and C. Crépeau, *Minimum Disclosure Proofs of Knowledge*, JCSS, v. 37, pp 156-189.

[BCY] Brassard G., C. Crépeau, and M. Yung, "Everything in NP can be proven in Perfect Zero Knowledge in a bounded number of rounds," *ICALP* 89. (Also in TCS).

[BY] Brassard G., and M. Yung, "One-Way Group Action," *CRYPTO* 90.

[CH] Chaum, D., "Demonstrating that a public predicate can be satisfied without revealing any information about how", Crypto 86.

[CDG] D. Chaum, I. Damgård and J. van de Graaf, *Multiparty Computations Ensuring Secrecy of each Party's Input and Correctness of the Output*, Proc. of Crypto 87, pp. 462.

[FFS] U. Feige, A. Fiat and A. Shamir, *Zero-Knowledge Proofs of Identity*, STOC 87, pp. 210-217.

[GILVZ] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan, and D. Zuckerman, *Security Preserving Amplification of Hardness*, FOCS 90.

[GKL] O. Goldreich, H. Krawczyk, and M. Luby, *On the Existence of Pseudo-Random Generators* , FOCS 88.

[GMW] S. Goldreich, S. Micali and A. Wigderson, *Proofs that Yields Nothing But their Validity*, FOCS 86, pp. 174-187. (also in JACM).

[GMR] S. Goldwasser, S. Micali and C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, STOC 85, pp. 291-304. (also in SIAM J. COMP.)

[H] Håstad, J., "Pseudo-Random Generators under Uniform Assumptions", *STOC 90*.

[ILL] I. Impagliazzo, L. Levin and M. Luby, *Pseudo-random generation from one-way functions*, Proc. 21st Symposium on Theory of Computing, 1989, pp. 12-24.

[IL] R. Impagliazzo and M. Luby, *One-way Functions are Essential for Complexity-Based Cryptography* FOCS 89, pp. 230-235.

[IN] R. Impagliazzo and M. Naor, *Efficient Cryptographic Schemes Provably as Secure as Subset-Sum*, Proc. of FOCS 89, pp. 236-241.

[IR] R. Impagliazzo and S. Rudich, *On the Limitations of certain One-Way Permutations* , Proc. of STOC 89, pp. 44-61.

[IY] R. Impagliazzo and M. Yung, *Direct Minimum-Knowledge Computations* , Proc. of Crypto 87, Springer Verlag.

[Kon] A. G. Konheim, *Cryptography: a primer* , Wiley, New York, 1981.

[N] M. Naor *"Bit Commitment Using Pseudo-Randomness"* Crypto-89 pp.123-132.

[NY] M. Naor and M. Yung, *Universal One-Way Hash Functions and their Cryptographic Applications*, STOC 89.

[OVY1] R. Ostrovsky, R. Venkatesan, M. Yung, *Fair Games Against an All-Powerful Adversary, Sequences 91*, (to appear).

[OVY2] R. Ostrovsky, R. Venkatesan, M. Yung, *Secure Commitment Against A Powerful Adversary, STACS 92*, Springer Verlag LNCS, 1992.

[OW] R. Ostrovsky and A. Wigderson, *One-Way Functions are Essential For Non-Trivial Zero-Knowledge Proofs*, (preliminary manuscript).

[Ro] J. Rompel "One-way Functions are Necessary and Sufficient for Secure Signatures" STOC 90.

[TW] M. Tompa and H. Woll, *Random Self-Reducibility and Zero-Knoweldge Interactive-Proofs of Possession of Information*, Proc. of FOCS 1987.

[Y82] A. C. Yao, *Theory and Applications of Trapdoor functions*, Proceedings of the 23th Symposium on the Foundation of Computer Science, 1982, pp 80-91.