Publicly Verifiable Non-Interactive Zero-Knowledge Proofs

Dror Lapidot Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel

Abstract

In this paper we construct the first publicly verifiable non-interactive zero-knowledge proof for any NP statement under the general assumption that one way permutations exist. If the prover is polynomially bounded then our scheme is based on the stronger assumption that trapdoor permutations exist. In both cases we assume that P and V have a common random string, and use it to prove a single theorem (which may be chosen as a function of the known string).

1 Introduction

The notion of a non-interactive zero-knowledge (NIZK) proof was introduced by [BlFeMi]. It allows a prover to prove in writing (without interaction) any NP-theorem to a polynomially bounded verifier, without revealing any knowledge besides the validity of the theorem, provided that they possess a common random string (such as the 1,000,000 random digits published by the RAND corporation). These NIZK proofs should be *publicly verifiable* (i.e. checkable by anyone rather than directed at a particular verifier) and zero-knowledge to any coalition of verifiers. Such proofs have important cryptographic applications, such as digital signatures, message authentication (see [BeGo]), and protection of public key cryptosystems against chosen ciphertext attacks (see[NaYu]).
[BlFeMi] and [DeMiPe] describe concrete implementations of this model based on the difficulty of specific computational problems (distinguishing products of two primes from products of three primes or distinguishing quadratic residues from quadratic non residues). Under the assumption that Oblivious Transfer protocols exist, [KiMiOs] and [BeMi] show how after an initial preprocessing stage, the prover can noninteractively prove polynomially many NP-statements, but these proofs are not publicly verifiable and all of them are directed to a particular verifier.
Finally the scheme of [DeMiPe1] and the preliminary scheme we present in section 2

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 353-365, 1991.

are based on a model in which the prover proves a random theorem in an interactive preprocessing stage and then uses it to prove the actual theorem noninteractively. These two schemes can be implemented using any one-way function.

Our main result in this paper is a publicly verifiable NIZK proof with a common random string, for any NP-theorem, under the general assumption that one-way permutations exist. The protocol remains zero-knowledge even when the theorem is chosen as a function of the random string. If the prover is polynomial time bounded, then our scheme is based on the stronger assumption that trapdoor permutations exist. This is the first known protocol of this type which is not based on the difficulty of specific computational problems.

Our result together with the result of [NaYu] imply that under the general assumption that trapdoor permutations exist, there exists a public key cryptosystem which is provably secure against chosen ciphertext attacks.

The paper is organized in the following way: In Section 2 we present a new construction of NIZK proofs with preprocessing which are as efficient as their interactive counterparts. In Section 3 we describe our main result and in Section 4 we prove its correctness. Section 5 is devoted to several extensions and applications of the main result.

2 A NIZK proof with preprocessing

Consider a prover who wants to prove the Hamiltonicity of an arbitrary graph G with n nodes. We assume that the prover and the verifier can execute a preliminary interactive stage which is independent of G (i.e. at this stage they know that in the non-interactive stage the prover will prove the Hamiltonicity of an n node G, but they don't know which graph it will be). Only after the termination of this interactive stage, they get G and execute the non-interactive move in which the prover sends a written message to the verifier in order to convince him in zero-knowledge that G is Hamiltonian. The verifier is not allowed to ask the prover any questions and should be convinced just by reading this message.

The Basic Step

Let H be a randomly chosen Hamiltonian cycle on n nodes. The adjacency matrix of H is a permutation matrix with a single 1 in each row and column, and a single cycle. Let S be such an adjacency matrix in which each entry is replaced by a string which hides it (for example: by the hard bit construction of [GoLe] or by a probabilistic encryption), so that a polynomially bounded observer cannot determine the locations of the 1's.

Assume now that S is given to P and V, and that P wants to prove to V the Hamiltonicity of some graph G with n nodes. Since P is infinitely powerful, he can recover the original Hamiltonian cycle H from S and determine the permutation π that maps H onto the Hamiltonian cycle of G (i.e., $\pi(H) \subseteq G$). To convince V that G is Hamiltonian, P just sends him (in writing) the permutation π and the original values of all the entries in $\pi(S)$ which do not correspond to edges in G. V

accepts the proof iff all the revealed entries are 0, since this implies that the n 1's that remain in $\pi(S)$ correspond to edges of G . The proof is zero knowledge since all the verifier gets is a random permutation and a collection of encryptions of 0's, which can be easily simulated.
The resulting NIZK proof with preprocessing (regardless of whether P is polynomially bounded or not) is executed as follows: In the preliminary interactive stage P sequentially sends k (=security parameter) such random matrices $S_1, S_2, \ldots, S_k$ to V and receives k random bits $b_1, b_2, \ldots, b_k$ from V. In the non-interactive move he reveals all the entries of those S_i's for which $b_i = 0$, and executes the basic step for those S_i for which $b_i = 1$. If all the S_i with $b_i = 0$ are of the appropriate form, V can conclude with high probability that at least one of the other S_i is also proper, in which case G is guaranteed to be Hamiltonian.
In order to compare this protocol to Blum's protocol for Hamiltonicity [Bl], lets recall that in the first move of Blum's scheme P randomly permutes G and sends V the encrypted adjacency matrix of this isomorphic copy. V then sends a random bit to P and according to that bit P either reveals all the entries in the matrix and the permutation, or reveals only the entries which correspond to the edges of the Hamiltonian cycle. Our protocol resembles Blum's protocol, with one major difference: In Blum's protocol all the moves depend on G, while in our protocol only the last move depends on G. As a result, Blum's protocol cannot be split into a preprocessing stage and a non-interactive proof as we did in our protocol.
Remark:
The NIZK proof with preprocessing can be extended to a variety of graph theoretic problems which are satisfied by a single minimal graph (under isomorphism). This family includes: Clique, Graph partition into triangles, Graph partition into cliques (and therefore also Graph coloring), 3-Dimensional Matching etc.

3 A NIZK Proof with A Common Random String

In this section we show that under the assumption that oneway permutations exist, if the prover and the verifier initially share a common random string then the initial preprocessing stage of our protocol can be discarded, yielding a NIZK proof for any NP statement in the original noninteractive model of Blum, Feldman and Micali.

3.1 Definitions

Definition: For any NP language L, Let R_L be the relation which contains all the pairs (x, ω) such that $x \in L$ and ω is a witness for that. $\diamond$
The input of P is a pair of words (x, ω) and the common random string σ whose length is polynomial in the size of x and in the security parameter k.
Notation: $A(x, y, z)$ denotes the output of a probabilistic algorithm A on input (x, y, z).
Definition: A non interactive proof system for an NP language L is a pair of

probabilistic algorithms (P, V) (where V is polynomially bounded) satisfying:

1. *Completeness:* $\forall(x,\omega) \in R_L,\ \forall\sigma\ \ V(x,\sigma,P(x,\omega,\sigma)) = accept$.
2. *Soundness :* If σ is a random string then the probability of succeeding in proving a false statement is negligible, even if the theorem is chosen by P after seeing σ. Formally:

$$\exists b\ \ \exists c\ \ \ \forall d\ \ \ \exists N\ s.t.\ \forall k > N$$

at least $(1 - \frac{1}{k^d})$ of the strings σ of length $|x|^b k^c$ satisfy:

$$\forall x' \notin L\ \ \forall y\ \ \ V(x',\sigma,y) = reject.$$

Definition: A non-interactive proof system for an NP-language L is zero-knowledge if there exists a random polynomial time simulator M such that for any $(x,\omega) \in R_L$, the two ensembles $(\sigma, P(x,\omega,\sigma))$ and $M(x)$ are polynomially indistinguishable (by nonuniform distinguishers). Formally:

$$\exists M\ s.t.\ \forall D\ \forall(x,\omega) \in R_L\ \forall d$$

$$|Pr(D(M(x)) = 1) - Pr(D(\sigma, P(x,\omega,\sigma)) = 1)| < \frac{1}{k^d}$$

for all sufficiently large k.
The probabilities are taken over the choices of σ and over the coin tosses of P and M.

3.2 Informal Description

Assume that P and V possess a common random string (CRS) and P wants to send V a non-interactive zero-knowledge proof based on the CRS, (rather than on an interactive preprocessing stage) that an arbitrary n node graph G is Hamiltonian. We do this by mapping the CRS into an appropriate sequence of matrices which contain with high probability at least one Hamiltonian matrix. P can then proceed exactly as in the final non-interactive step of the protocol described in Section 2.
How can P construct such matrices? It is possible to get a sequence of hidden random bits from the CRS by calculating an appropriate hard bit of a one-way permutation with respect to each segment of it. But if we naively pack such a block of n^2 hidden random bits into a $n \times n$ 0/1 matrix, the probability that this is a Hamiltonian matrix is exponentially small. Therefore in order to solve this problem we have to transform the CRS into a matrix in a more complicated way.
Assume that the CRS defines a $n^2 \times n^2$ matrix B of zeroes and ones, such that $Pr\{B_{i,j} = 1\} = 1/n^3$ for each (i,j) and this matrix has the same security properties as S. In order to construct a matrix such as S from a given matrix B and to prove that G is Hamiltonian P has to execute the following:

1. If the number of 1's in B is different from n or there exists a row or a column which contains at least two 1's, then P proves this fact by revealing all the entries in B.

2. Otherwise (i.e. B contains a $n \times n$ permutation submatrix), P reveals to V all the entries in the $n^2 - n$ rows and the $n^2 - n$ columns which contain only zeroes, and removes them from B. If the resulting $n \times n$ matrix does not represent a single cycle, P proves this fact to V by revealing all the entries of the remaining matrix.

3. Otherwise (i.e. the remaining matrix represents a single cycle), the original matrix B is called good and P must use the resulting $n \times n$ matrix in the execution of the protocol described in the previous section.

What's left is to show how to transform the CRS into B and to prove that such a matrix is good with sufficiently high probability.
Consider the CRS as a concatenation of polynomially many blocks of k random bits. Let f be a one way permutation that both P and V can evaluate but only P can invert. [GoLe] prove the existence of a hard bit in any one way function. Therefore if we associate such a hard bit with each block of the CRS, we get a new hidden random string (HRS). More precisely, let r' and r" be two consecutive blocks of k bits in the CRS, let $x = f^{-1}(r')$ and $y = r''$ and let s be the hidden random bit defined by the scalar product of the boolean vectors x, y. This process transforms the sequence of blocks in the CRS into a sequence of hidden random bits.
All we have to show is how to transform the HRS into a sequence of matrices such as B. Consider the HRS as a concatenation of polynomially many consecutive blocks of m bits where $m = log(n^3)$ (w.l.g. we can assume that it is an integer). We interpret a block as 1 if all its m bits are 1 and 0 otherwise, and thus we can pack each consecutive segment of n^4m hidden random bits into the desired $n^2 \times n^2$ 0/1 matrix B discussed above. In Section 4.2 we prove that the probability that such a matrix is good is $\frac{1}{poly(n)}$, and therefore if the length of the CRS is large enough (polynomial in k and n) then with high probability at least one of the segments defines a Hamiltonian matrix for which P must executes the basic step described in section 2.
In order to formally describe the scheme (which is slightly more efficient than the informal scheme described above) and prove its correctness we introduce some notations and definitions.

3.3 Notations and Definitions

Let $r_1 \circ r_2 \circ \ldots \circ r_{poly(k,n)}$ (where $r_l \in_R \{0,1\}$ for each l, and $\circ$ denotes concatenation) be the common random string (CRS), shared by P and V. Let f be a one-way permutation whose definition is known to both of them. Let $u_1 \circ u_2 \circ \ldots \circ u_{poly(k,n)}$ (where $u_i \in \{0,1\}$ for each i) be an intermediate random string (IRS) which is

defined as follows: For each $j \geq 1$,

$$f(x_{j,1}) = y_{j,1} \quad and \quad x_{j,2} = y_{j,2}$$

where:

$$x_{j,1} = u_{2k(j-1)+1} \circ u_{2k(j-1)+2} \circ \ldots \circ u_{2k(j-1)+k}$$

$$x_{j,2} = u_{2k(j-1)+k+1} \circ u_{2k(j-1)+k+2} \circ \ldots \circ u_{2kj}$$

$$y_{j,1} = r_{2k(j-1)+1} \circ r_{2k(j-1)+2} \circ \ldots \circ r_{2k(j-1)+k}$$

$$y_{j,2} = r_{2k(j-1)+k+1} \circ r_{2k(j-1)+k+2} \circ \ldots \circ r_{2kj}.$$

Let $s_1 \circ s_2 \circ \ldots \circ s_{poly(k,n)/2k}$ be the hidden random string (HRS) which is defined as follows: for each $j \geq 1$, s_j is the scalar product of the boolean vectors $x_{j,1}$ and $x_{j,2}$. This construction is based the theorem of [GoLe] which says that, according to these notations, given random $y_{j,1}$ and $y_{j,2}$, s_j is a hard bit.
For each $i \geq 1$ let a_i be such that its binary representation is $s_{(i-1)m+1} \circ s_{(i-1)m+2} \circ \ldots \circ s_{im}$. Lets define for each i:

$$b_i = \begin{cases} 1 & \text{if } a_i = 2^m - 1 \\ 0 & \text{otherwise} \end{cases}$$

Let B_i be a $n^2 \times n^2$ matrix which is defined as follows: $B_i(j,l) = b_{(i-1)n^4+(j-1)n^2+l}$ for every $1 \leq i, j, l$.
Definition: We say that B_i is a *proper* matrix if it contains exactly n ones and each column and row contains at most a single one.
If B_i is a proper matrix let N_i be the $n \times n$ matrix obtained by removing all the n^2-n columns and n^2-n rows which contain only zeroes. Otherwise N_i is undefined.
Definition: We say that N_i is a Hamiltonian matrix if there is a permutation $\psi \in S_n$ with a single cycle such that for each $N_i(l,j)$ which is equal to 1 $j = \psi(l)$. In this case we say that B_i is a good matrix.

3.4 The Scheme

Assume that P and V have a CRS with $2n^7km$ bits and a common one-way permutation f.
P's protocol:
For each $1 \leq i \leq n^3$ do the following:

1. If B_i contains more than n ones then reveal $n+1$ of them.
2. If B_i contains fewer than n ones then reveal all the entries.
3. If there is a column or row which contains two ones then reveal the two entries.
4. (B_i is a proper matrix) Reveal and remove all the $n^2 - n$ columns and all the $n^2 - n$ rows which contain only zeroes. If N_i is not a Hamiltonian matrix then reveal the n ones. Otherwise use N_i in the execution of the protocol described in section 2.

V's protocol:
For each $1 \leq i \leq n^3$ do the following:

1. If P reveals $n+1$ entries then check that all of them are 1.
2. If P reveals all the entries then check that B_i contains fewer than n ones.
3. If P reveals two entries then check that both of them are 1 and in the same column or row.
4. If P reveals $n^2 - n$ columns and $n^2 - n$ rows then check that all the entries in these rows and columns are zeroes.
5. If P reveals n entries then check that all of them are 1 and N_i is not a Hamiltonian matrix.
6. Otherwise check that the protocol described in Section 2 is carried out correctly.

Accept the proof iff for each $1 \leq i \leq n^3$ one of these checks is successful.

3.5 NIZK Proof for Some Other NP-Statements

The same technique can be used (without reductions) to prove other NP-Complete statements. Consider for example the 3-Dimensional Matching ($3DM$) problem. Each instance of the problem is a 3-dimensional 0/1 matrix M ($n \times n \times n$) and P's goal is to prove that there are n ones in M such that no two of them agree in any coordinate.
Consider each block in the CRS as a hidden random 3-dimensional 0/1 matrix whose size is $n^2 \times n^2 \times n^2$ and set the probability of 1 at each entry to $1/n^5$. The same proof technique implies that with high probability there is a block in the CRS which hides a good matrix B, namely a matrix with exactly n ones such that no two of them agree in any coordinate. P reveals all the 2-dimensional submatrices of B which contain only zeroes so that the remaining $n \times n \times n$ hidden matrix N forms a random minimal example for 3-dimensional matching.
To prove that a given M contains a $3D$ matching, P sends to V the permutation that moves the n ones in N to the locations of the matching in M, and then proves that every 0 in M corresponds to a zero in the permuted N.

4 Correctness

4.1 Completeness

The non-interactive proof of Hamiltonicity is complete because in every $n^2 \times n^2$ matrix that does not yield a Hamiltonian matrix, all P has to do is to open some of its entries, and V will accept his proof as valid.

4.2 Soundness

Lemma : The probability that B_i contains exactly n 1's is $\geq 1/3n$, for every i.
Proof : The bits of the HRS are unbiased and independent, and for each j the probability that $b_j = 1$ is $1/n^3$. Therefore the expected number of 1's in B_i is n. If x denotes the number of 1's then Chebyshev's Inequality implies that

$$Pr\{|x - n| > n\} < \frac{Var(x)}{n^2} = \frac{n^4 n^{-3}(1 - n^{-3})}{n^2} < n^{-1}$$

therefore

$$\sum_{i=0}^{2n} Pr\{x = i\} > 1 - n^{-1}.$$

Since the maximal probability is at $x = n$

$$Pr\{x = n\} > \frac{1 - n^{-1}}{2n + 1} > 1/3n \qquad \square$$

The size of B_i is $n^2 \times n^2$ and therefore by the birthday paradox if B_i contains exactly n 1's then the probability that each row and each column contains at most one 1, is a constant.
The number of permutations in S_n which consist of a single cycle (of length n) is $(n-1)!$, therefore the probability that N_i is a Hamiltonian matrix, given that it is a permutation matrix, is n^{-1}.
We conclude that, for every i, the probability that B_i yields a Hamiltonian matrix N_i is $\geq dn^{-2}$, where d is a constant. Thus if the length of the CRS is $O(n^7 km)$ bits then with probability $(1 - e^{-n})$ at least one of the B_i's yields a Hamiltonian matrix. Any such matrix will expose a cheating P.
Remark: If $log(n^3)$ is not an integer, we have to set $m = \lceil log(n^3) \rceil$ and choose B_i as a $\lceil bn^2 \rceil \times n^2$ matrix where $b = \frac{2^m}{n^3}$ $(1 < b < 2)$.

4.3 Zero-Knowledge

In order to simplify the proof of zero-knowledge we refer only to the informal scheme described in (3.1). We construct a random polynomial time simulator M which generates a "random string" and a "proof" of Hamiltonicity which are polynomially indistinguishable (by nonuniform distinguishers) from those generated by a real execution of the protocol.
We use the transitivity of the property of indistinguishability: First we construct a random polynomial time algorithm P' (with access to the Hamiltonian cycle of G) whose output is indistinguishable from a truly random string appended to a proof of the real prover, and then we construct a random polynomial time simulator M (who does not know the Hamiltonian cycle) whose output is polynomially indistinguishable from that of P'. Therefore these constructions imply that our scheme is zero-knowledge.

Let P' be the random polynomial time algorithm which executes the real protocol with the following exception: it chooses a sequence of truly random bits (IRS), and then gets the CRS by applying the one-way permutation f in the forward direction. Clearly the output of P' is indistinguishable from that of the real prover.
The simulator M accepts G and the security parameter k as inputs, and outputs a string σ_k of length $2n^7km$ bits and a "proof" in the following way:

1. M randomly chooses a sequence of $2n^7km$ truly random bits and uses them as the intermediate random string (IRS). In every segment that yields a Hamiltonian matrix it randomly changes the interpretation of all the ones to zeroes. More precisely: For each i for which N_i is a Hamiltonian matrix and for each j,l such that $N_i(j,l)=1$, M randomly and independently chooses $2km$ bits instead of: $u_{((i-1)n^4+(j-1)n^2+(l-1))m2k+1}\cdots u_{((i-1)n^4+(j-1)n^2+l)m2k}$ until $N_i(j,l)=0$ (the probability of success is $1-\frac{1}{n^3}$).

2. M transforms the modified IRS into a common random string (CRS) σ_k by applying f in the forward direction and computes the [GoLe] hidden random string (HRS) as the dot product of consecutive pairs of blocks in the IRS.

3. For each i such that B_i has not changed in the first step M reveals all the entries of B_i. For each of the other B_i's it randomly reveals n^2-n rows and n^2-n columns. Since the resulting $n\times n$ matrix contains only zeroes, M can easily simulate the basic step by choosing a random permutation $\psi\in_R S_n$ and revealing every $B_i(j,l)$ such that there is no edge between j and l in $\psi(G)$.

The output of M is denoted by $(\sigma_k, proof'(\sigma_k,G))$ where the second component includes all the revealed bits and permutations. Let τ_k be a string of length $2n^7km$ bit, and denote by $proof(\tau_k,G)$ a proof of P' based on G and τ_k.
For any nonuniform distinguisher D, let $D(x)$ denote the 0/1 output of D on input x. Let

$$P_{P,k}=Pr\{D((\tau_k,proof(\tau_k,G)),G)=1\}$$
$$P_{M,k}=Pr\{D((\sigma_k,proof'(\sigma_k,G)),G)=1\}.$$

The probabilities are taken over the choices of τ_k and over the coin tosses of P' and M.
Theorem: For any Hamiltonian graph G , for any nonuniform random polynomial time distinguisher D and for any polynomial Q :

$$|P_{P,k}-P_{M,k}|<\frac{1}{Q(k)}$$

for all sufficiently large k.
Proof: Assume that there exists an efficient distinguisher D, a polynomial Q and an infinite subset $\mathcal{I}\subset\mathcal{N}$ such that for every $k\in\mathcal{I}$:

$$(*)\quad |P_{P,k}-P_{M,k}|\geq\frac{1}{Q(k)}.$$

Let k be an element in $\mathcal{I}$. Let $\alpha = (i_1, \ldots, i_t, \psi_1, \ldots, \psi_u)$ $(1 \leq i_1 < \ldots < i_t \leq n^7 m$ and for each $1 \leq i \leq u$ $\psi_i \in S_n)$ and let $P_{\alpha,k}$ $(P'_{\alpha,k})$ be the probability that $s_{i_1}, \ldots, s_{i_t}$ are the hidden bits revealed by P' (M) and $\psi_1, \ldots, \psi_u$ are the permutations given by P' (M) (each one associated with a Hamiltonian matrix). Since τ_k is a truly random string, M simulates P' and all the choices of M are random we conclude that for any α:

$$P_{\alpha,k} = P'_{\alpha,k}.$$

Let $proof(\tau_k, G, \alpha)$ and $proof'(\sigma_k, G, \alpha)$ denote proofs of P' and M based on τ_k and σ_k respectively, in which the revealed bits and the random permutations are according to α. It is obvious that in the case of P', once τ is chosen, α is fixed.
Denote by $P_{P,\alpha,k}$ the probability that D outputs 1 on the input $(\tau_k, proof(\tau_k, G, \alpha))$ (while τ is a truly random string) and by $P_{M,\alpha,k}$ the probability that D outputs 1 on input $(\sigma_k, proof'(\sigma_k, G, \alpha))$.
It is obvious that

$$(**) \quad P_{P,k} = \sum_{\alpha} P_{\alpha,k} P_{P,\alpha,k}$$

and

$$(***) \quad P_{M,k} = \sum_{\alpha} P_{\alpha,k} P_{M,\alpha,k}.$$

The following Lemma claims that for any α, D is unable to distinguish between $(\tau_k, proof(\tau_k, G, \alpha))$ and $(\sigma_k, proof'(\sigma_k, G, \alpha))$.
Lemma: For every α

$$|P_{P,\alpha,k} - P_{M,\alpha,k}| < \frac{1}{Q(k)}.$$

Proof: Assume that this is not true, namely there is α for which w.l.g.

$$P_{P,\alpha,k} - P_{M,\alpha,k} \geq \frac{1}{Q(k)}.$$

For every $1 \leq j \leq n^7 m$, $P^j_{P/M,\alpha,k}$ denotes the probability that D outputs 1 on the following (string, proof): The first $2k(j-1)$ bits in the string are randomly chosen (a prefix of a real CRS) and associated with a proof of P' until that point, while all the other bits and the rest of the proof are generated by M and both of these parts follow the vector α. Following the well known Hybrid argument of [GoMi] we conclude that there is $1 \leq i \leq n^7 m$ for which :

$$P^{i+1}_{P/M,\alpha,k} - P^i_{P/M,\alpha,k} \geq \frac{1}{Q(k) n^7 m} \quad .$$

From the description of P' and M we conclude that i is the index of one of the hidden bits of one of the appearances of $\underbrace{1,1,\ldots,1}_{m}$ in a segment which defines a Hamiltonian matrix in the simulation of P'. We'll construct a random polynomial time nonuniform algorithm C_k whose auxiliary input is the graph G, including the definition of a Hamiltonian cycle, α and i which on input $(f(x), y)$ $(x, y$ are randomly

chosen) outputs a bit b which is the hard bit of $(f(x), y)$ with probability $\geq \frac{1}{2} + \frac{1}{poly(k)}$. This is a contradiction to the assumption that f is one-way. This algorithm uses P', M and D as subroutines and executes the following steps:

1. Run P' so that the indices of the hidden bits which are revealed and the permutations associated with the Hamiltonian matrices are according to α.
2. Run M according to the same rule.
3. Erase from the output of P' all the bits coming after the $(i-1)$'th block, namely remain with the first $2(i-1)k$ bits of the string appending the revealed bits and the permutations associated with the Hamiltonian matrices (call this prefix S_P).
4. Erase from the output of M the first i blocks, namely remain with the last $2n^7km - 2ik$ bits of the string appending the revealed bits and the permutations associated with the Hamiltonian matrices (call this suffix S_M).
5. Feed D with $S_P \circ f(x) \circ y \circ S_M$.
6. If $D(S_P \circ f(x) \circ y \circ S_M) = 1$ then $b = 1$ else $b = 0$.

It is easy to verify that with probability $\geq \frac{1}{2} + \frac{1}{poly(k)}$, b is the hard bit of $(f(x), y)$ and this is a contradiction to the assumption that f is one-way. □
This lemma together with $(**)$ and $(***)$ contradicts $(*)$ which completes the proof of the theorem. □

Remark: Consider an NP-statement which is polynomially chosen as a function of the random string namely, there is a nonuniform random polynomial time algorithm which gets a random string and outputs an NP-statement (which is a function of it) including an appropriate witness.
The simulator generates the "random string" independently of the NP-theorem. Therefore considering the construction of the appropriate C_k, we conclude that:

Corollary: Our non-interactive proof remains zero-knowledge even if the NP-statement (of size n) is polynomially chosen as a function of the common random string.

5 Extensions and Applications

5.1 A Polynomial Time Prover

If the prover is polynomial time bounded then our scheme is based on the stronger assumption that trapdoor permutations exist. In fact, we assume that for every security parameter there exists an exponentially large family of trapdoor permutations whose indices are n^c bit strings (c is constant). The only difference from the scheme described in section 3 is that now P randomly chooses a trapdoor permutation f from that family, sends its index to V and keeps the trapdoor information

secret. Now the ability of P to invert f is implied by his knowledge of the trapdoor. The proof of completeness remains unchanged, but there might be a problem with the soundness: In contrast to the scheme described in section 3 in which the (unbounded) prover does not choose the one-way permutation, in this scheme a cheating prover may choose a particularly useful trapdoor permutation after seeing the CRS. To overcome this difficulty, we only have to extend the CRS: If the number of bits in it is $O(n^{6+c}km)$ then the probability of cheating in our scheme is at most $O(\frac{2^{n^c}}{e^{n^c}})$ since this is an upper bound on the fraction of random strings which can be bad for any trapdoor permutation.
The proof of the zero-knowledge property resembles its counterpart for the original scheme, except that we have to consider all the choices of trapdoor permutations.

5.2 Public-Key Cryptosystems Secure against Chosen Ciphertext Attacks

The existence of public-key cryptosystems which are secure against passive eavesdropping under the assumption that trapdoor permutations exist is well known. [NaYu] show how to construct a public-key cryptosystem which is provably secure against chosen ciphertext attacks (CCS-PKC), given a public-key cryptosystem which is secure against passive eavesdropping and a non-interactive zero-knowledge proof system in the shared string model. Using their result together with our construction (for polynomial time provers) we have:
Corollary: CCS-PKC exist under the general assumption that trapdoor permutations exist.
This is the first known CCS-PKC which is not based on the difficulty of specific computational problems.

5.3 Multiple NIZK Proofs

We have to emphasize that our scheme is a bounded NIZK proof system in the sense that using a random string, the prover can prove in zero-knowledge only a single theorem. Recently, Feige, Lapidot and Shamir [FeLaSh] have shown how to transform any bounded NIZK proof system with polynomial time provers into a general NIZK proof system in which polynomially many independent provers can share the same random string and use it to prove polynomially many statements of polynomial length in a completely memoryless way.

References

[BeGo] M. Bellare, and S. Goldwasser, *New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero-Knowledge Proofs.* CRYPTO 89.

[BeMi] M. Bellare, and S. Micali, *Non-Interactive Oblivious Transfer and Applications.* CRYPTO 89.

[Bl] M. Blum, a presentation in the 1986 International Congress of Mathematics.

[BDMP] M.Blum, A. De Santis, S. Micali and G. Persiano, *Non-Interactive Zero-Knowledge* , Manuscript, December 1989.

[BlFeMi] M. Blum, P. Feldman, and S. Micali, *Non-Interactive Zero-Knowledge Proof Systems and Applications*, Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 1988.

[DeMiPe] A. De Santis, S. Micali, and G. Persiano, *Non-Interactive Zero-Knowledge Proof Systems*, CRYPTO 87.

[DeMiPe1] A. De Santis, S. Micali, and G. Persiano, *Non-Interactive Zero-Knowledge with Preprocessing*, CRYPTO 88.

[FeLaSh] U. Feige, D. Lapidot and A. Shamir, *Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String.* FOCS 90.

[FeSh] U. Feige, and A. Shamir, *Zero-Knowledge Proofs of Knowledge in Two Rounds.* CRYPTO 89.

[GoLe] O. Goldreich, L.A. Levin, *A Hard-Core Predicate for all One-Way Functions.* STOC 89.

[GoMiWi] O. Goldreich, S. Micali, and A. Wigderson, *Proofs that Yield Nothing But Their Validity and a Methodology of Cryptographic Protocol Design* . Proc. 27th FOCS, 1986.

[GoMi] S. Goldwasser, and S. Micali *Probabilistic Encryption*, Journal of Computer and System Sciences 28, 1984

[GoMiRa] S. Goldwasser, S. Micali, and C. Rackoff, *The Knowledge Complexity of Interactive Proofs.* STOC 85.

[KiMiOs] J. Kilian, S. Micali, and R. Ostrovsky, *Minimum Resource Zero-Knowledge Proofs.*, FOCS 89.

[NaYu] M. Naor, M.Yung, *Public-key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks* , Proc. of STOC 90.