Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String

Extended Abstract

Uriel Feige, Dror Lapidot and Adi Shamir
Dept. of Applied Math.
The Weizmann Institute
Rehovot 76100
Israel

Abstract

In this paper we solve the two major open problems associated with non-interactive zero knowledge proofs, as stated by Blum, De Santis, Micali and Persiano: How to enable polynomially many provers to prove in writing polynomially many theorems based on a single random string, and how to construct such proofs under general (rather than number-theoretic) assumptions. Our new constructions can be used in cryptographic applications in which the prover is restricted to polynomial time, and are much simpler than earlier (and less capable) proposals.

1 Introduction

1.1 Background

Blum, Feldman, and Micali [BFM88] suggested the intriguing concept of *noninteractive zero knowledge* (NIZK), aimed at eliminating the interaction between prover and verifier in zero knowledge interactive proof systems [GMR85]. The prover P writes down a zero knowledge proof that an input x belongs to a prespecified language L, and any verifier V can check the validity of this written proof against a universal publicly available random string (such as the RAND string of one million random digits), called the *reference string*. NIZK has become an important primitive for cryptographic protocols, with applications such as signature schemes [BG89] and encryption schemes secure against chosen ciphertext attack [NY90].

NIZK proof systems for any NP statement were constructed in [BFM88] and [DMP87], under a specific number theoretic assumption. The main disadvantage of these *bounded* NIZK proofs is that the prover can prove only one statement of size bounded by the length of the reference string: If polynomially many proofs are given using the same reference string, the zero knowledge property breaks down[1]. In [BDMP89] it was finally shown how a single prover can give polynomially many proofs using the same reference string, but the scheme is still based on a specific number theoretic assumption, and cannot support polynomially many provers.

Not knowing whether NIZK protocols can be constructed under general (rather than number theoretic) cryptographic assumptions, De Santis, Micali and Persiano [DMP88] suggested a slight variation of the NIZK model. In their *noninteractive with preprocessing* model, the verifier and prover create a common reference string (which need not look like a random string) during an interactive preliminary stage. Based on this reference string, the prover can then prove any single NP statement (of bounded length). Unlike the original NIZK model, in the preprocessing model the proof should look convincing only to the verifier who takes part in the initial preprocessing stage, which makes this model unsuitable for applications such as signature schemes. [DMP88] showed an implementation of this idea based on the general assumption that oneway functions exist. Under the stronger cryptographic assumption that *oblivious transfer* protocols exist, [KMO89] show how after an initial preprocessing stage, the prover can noninteractively prove polynomially many NP statements, but again the proof is verifiable only by its original recipient.

1.2 Our Results

In this paper we answer the two major open questions associated with the concept of non-interactive zero knowledge, as presented by Blum, De Santis, Micali and Persiano [BDMP89]: How to construct NIZK proof systems for any NP statement under general (rather than number theoretic) assumptions, and

[1] The method suggested in [BFM88] for overcoming this difficulty was found to be flawed.

CH2925-6/90/0000/0308$01.00 © 1990 IEEE

how to enable polynomially many provers to share the same random reference string in giving such proofs.

As a preliminary result leading to our solution of the first open question, we construct (under the assumption that oneway functions exist) a very simple zero knowledge *noninteractive with preprocessing* proof for Hamiltonicity, which is as efficient as the interactive one presented by Blum [Blum86]. In contrast, all the previously known constructions of *NIZK with preprocessing* proofs [DMP88] are more complex and less efficient than their interactive counterparts. Then, under the assumption that oneway permutations exist, we show that if the prover and 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. This noninteractive protocol is the only known implementation which relies on general cryptographic assumptions, and is conceptually simpler than the number-theoretic protocols presented by Blum, De Santis, Feldman, Micali and Persiano. Under the assumption that trapdoor permutations exist, our NIZK protocol can be carried out by probabilistic polynomial time provers, and thus can be used in cryptographic applications which require NIZK protocols.

As a solution to the second open problem, we show how to transform any bounded NIZK proof system for an NP complete language into a *general* NIZK proof system in which polynomially many independent provers can share the same reference string and use it to prove polynomially many statements of polynomial length. The transformation is based on the general assumption that oneway functions exist. Thus any weakening in the cryptographic assumptions made in the construction of bounded NIZKs is automatically reflected in the cryptographic assumptions made in the construction of general NIZKs.

Independently of our work, De Santis and Yung [DY] also show how to transform bounded NIZK proof systems into general ones, though their transformation produces noninteractive proofs which are longer than ours by several orders of magnitude.

Several authors observed that in order to use NIZK proof systems in cryptographic applications it is often necessary to extend the security conditions imposed on NIZKs to withstand adaptive attacks ([BG89], [NY90]). The original definitions of NIZK proof systems assume that the statements to be proved are chosen independently of the reference string, whereas the adaptive setting allows for the possibility that statements to be proven are chosen after the reference string is given, and may depend upon the reference string. In the last section of this paper we show that our constructions also satisfy the more stringent conditions imposed by the adaptive setting.

Remark: A preliminary version of our paper, which includes the results of Section 2, was presented at Crypto 90 [LS90].

1.3 Definitions

$A(x)$ denotes the output of a probabilistic algorithm A on input x. This is a random variable. $\nu(n)$ denotes any function vanishing faster than the inverse of any polynomial. Formally:

$$\forall k\ \exists N\ s.t.\ \forall n > N\ \ 0 \leq \nu(n) < \frac{1}{n^k}$$

Definition 1.1 *Let R be a relation $\{(x,w)\}$ testable in polynomial time, where the lengths of x and w are polynomially related. For any x, its* witness set $w(x)$ *is the set of w such that $(x,w) \in R$.*

Clearly, $L_R = \{x \mid \exists w\ s.t.\ (x,w) \in R\}$ is an NP language.

In a *noninteractive zero knowledge proof system* for L_R, the prover P and verifier V share a random reference string which is denoted by σ. The prover's goal is to write down a proof that a common input x of length n belongs to L_R. The verifier may check the validity of this proof against the reference string σ. The verifier is probabilistic polynomial time. Consequently, only a polynomial size prefix of σ is used. The prover has an auxiliary input w, which is a witness to the NP statement $x \in L_R$, and can be used by P in order to compute the noninteractive proof.

Definition 1.2 *A* noninteractive proof system *for an NP language L_R characterized by a relation R is a pair of algorithms (P,V), where V is probabilistic polynomial time, satisfying:*

1. Completeness: $\forall (x,w) \in R$
   $\text{Prob}(V(x,\sigma,P(x,w,\sigma))\ \text{accepts}) > 1 - \nu(n)$
2. Soundness: *Even a cheating prover P' cannot convince V to falsely accept x on a randomly chosen σ. Formally:*
   $\forall P'\ \forall x \notin L_R\ \forall w'$
   $\text{Prob}(V(x,\sigma,P'(x,w',\sigma))\ \text{accepts}) < \nu(n)$

The probabilities are taken over the choices of σ, and over the coin tosses of P and V.

Definition 1.3 *A noninteractive proof system for relation R is* bounded zero knowledge *if there exists a random polynomial time simulator M such that for*

any $(x,w) \in R$, the two ensembles $(x,\sigma,P(x,w,\sigma))$ and $M(x)$ are polynomially indistinguishable (by nonuniform distinguishers). Formally:

$\exists M$ s.t. $\forall D\ \forall (x,w) \in R$

$|\text{Prob}(D(M(x)) = 1)$

$-\text{Prob}(D(x,\sigma,P(x,w,\sigma)) = 1)| < \nu(n)$

The probabilities are taken over the choices of σ, and over the coin tosses of P and M.

2 NIZKs Under General Cryptographic Assumptions

2.1 Preliminaries: Hidden Bits and the Basic Protocol

The concept of *bit commitment* is central to zero knowledge proofs. In an initial preprocessing stage, the prover and verifier agree on a common reference word which encodes a *hidden bit*. The prover can "open" the hidden bit in a unique way (either as 0 or 1), whereas the polynomial time bounded verifier cannot predict it with better than chance probability. As demonstrated by Naor [Naor89] (in conjunction with the results of [ILL89], [H90]), the existence of oneway functions is sufficient for the construction of bit commitments (and thus of hidden bits).

In this section we use the concept of hidden bits to construct zero knowledge proofs of Hamiltonicity of directed graphs. Since this is an NP complete property, our protocols can be used to prove any NP statement, via a reduction to an instance of Hamiltonicity.

We first present a preliminary protocol, on which we later base our NIZK proofs. 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 an encoding of such a matrix whose hidden bits can be read by the prover but not by the verifier. Assume now that P wants to use S in order to prove to V the Hamiltonicity of some graph G with n nodes.

Protocol 2.4 *Let π be a permutation that maps H onto the Hamiltonian cycle of G (i.e., $\pi(H) \subseteq G$). P sends V (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.

We explain informally why Protocol 2.4 is a zero knowledge proof of Hamiltonicity.

Completeness: P, which is either infinitely powerful or polynomial time with knowledge of a Hamiltonian cycle in G, can determine π and perform the protocol.

Soundness: P's proof implies that the n 1's that remain unopened in $\pi(S)$ correspond to edges of G, and thus G contains a Hamiltonian cycle.

Zero Knowledge: All the verifier gets is a collection of encodings of hidden bits some of which are opened as '0' and a random permutation, and both things can be simulated in random polynomial time.

2.2 NIZK With Preprocessing

Protocol 2.4 trivially gives a *NIZK with preprocessing* proof of Hamiltonicity. In the preliminary interactive stage (i.e., before the matrix G is known), P sequentially sends n encodings of matrices $S_1, S_2, \ldots, S_n$, each containing a different random Hamiltonian cycle. After each matrix S_i is received, V replies with a random bit b_i. This completes the preprocessing stage. In the non-interactive stage (after G is given), P reveals all the entries of those S_i's for which $b_i = 0$, and executes Protocol 2.4 for those S_i for which $b_i = 1$. If all the S_i with $b_i = 0$ contain Hamiltonian cycles, V can conclude with high probability that the same holds for at least one of the other S_i, and thus G is Hamiltonian.

It is instructive to compare our protocol with Blum's protocol for Hamiltonicity [Blum86]. 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 noninteractive protocol is just as efficient as Blum's interactive 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.

2.3 NIZK With a Common Random String

The common input is an n node directed graph G, and P wants to prove noninteractively that G contains a Hamiltonian cycle. In subsection 2.2 we showed how a preprocessing stage can be used to construct a matrix of hidden bits which contains a Hamiltonian cycle, and then Protocol 2.4 is used to prove G's Hamiltonicity. In this subsection we show that if P and V share a common random string, then with overwhelming probability they are already in the position where they possess a matrix with a hidden

Hamiltonian cycle, and the basic protocol can be executed without a prior interactive preprocessing stage.

We first note that the existence of oneway permutations suffices in order to interpret any *random* string c of length n as a hidden bit, provided the prover is exponential time. The prover can open the hidden bit by revealing the single preimage of c under f, and the hidden bit would be one of the "hard bits" of f (e.g., the inner product of the preimage with another prespecified random string [GL89]). Thus by sharing a random string, P and V share a sequence of bits which are hidden from V. This sequence can be divided into *blocks*, where each block represents an $n \times n$ hidden matrix, in such a way that with overwhelming probability, at least one of the blocks corresponds to a random Hamiltonian cycle.

The naive approach of using blocks of n^2 hidden bits and interpreting them as $n \times n$ 0/1 matrices fails, since the probability that such a matrix corresponds to a Hamiltonian cycle is exponentially small. Therefore, one has to construct the blocks in a more complicated way. There are many ways of doing this, and we describe just one of them.

We construct n^3 blocks. Block B_l, for each $1 \leq l \leq n^3$, is composed of an $n^2 \times n^2$ matrix. Each entry of each matrix is composed of $\log n^3 = 3 \log n$ consecutive hidden bits, which are interpreted as *edge* if all of them are '1', and otherwise they are interpreted as *nonedge*. Thus the probability that each entry is interpreted as *edge* is n^{-3} independently of other entries. For each block, the prover does the following:

1. If the number of *edge* entries is different from n, or if there exists either a row or a column which contains at least two *edges*, then P proves this fact by revealing all the entries in B.
2. Otherwise (i.e., the resulting $n \times n$ matrix represents a permutation), P reveals all the entries in the $n^2 - n$ rows and the $n^2 - n$ columns which contain only *nonedges*, and removes them from B. If the $n \times n$ matrix does not consist of a single cycle, P proves this fact to V by revealing all the remaining entries of the block.
3. Otherwise (i.e., the matrix represents a single cycle), P uses the resulting $n \times n$ matrix to prove the Hamiltonicity of the common input graph G by Protocol 2.4.

V accepts iff for each block P performs one of the above.

Theorem 2.5 *Under the assumption that oneway permutations exist, the above protocol is a noninteractive zero knowledge proof of Hamiltonicity.*

Proof (sketch):

Completeness: If G is Hamiltonian, P can obviously perform the protocol. (This is true even in the unlikely event that none of the blocks corresponds to a Hamiltonian cycle, since in this case all P has to do is to open all the hidden bits, and V will accept.)

Soundness: It is sufficient that at least one of the blocks corresponds to a Hamiltonian cycle, since on such a block P must apply Protocol 2.4, and it is impossible to do so successfully if the input graph is nonHamiltonian. Consider any block B_l. The probability that it contains exactly n *edge* entries is $\Omega(1/n)$ (standard probability theory). The probability that no two of them share the same column or the same row is constant (the "birthday paradox"). The probability that a random permutation is a single cycle is exactly $1/n$. Thus the probability that any single block corresponds to a Hamiltonian cycle is $\Omega(n^{-2})$ independently of other blocks. Since there are n^3 blocks, then with overwhelming probability at least one of them is good.

Zero-Knowledge: We construct a random polynomial time simulator M which, without knowledge of a cycle in G, generates a reference string and a "proof" of G's Hamiltonicity which are polynomially indistinguishable from the proof generated by the real prover.

Given a random reference string, polynomial time M cannot invert the oneway permutation f and recover its associated hard bits. Thus M's first problem is to obtain a random reference string σ for which he can reveal the hidden bits. M solves this problem by first creating a sequence of truly random bits, and then encoding them as hidden random bits by applying f in the forward direction on random arguments whose hard bits equal the desired hidden bits.

M partitions σ into blocks as P does, and opens all the blocks which do not correspond to a Hamiltonian cycle. Not knowing a Hamiltonian cycle in G, machine M encounters its second problem: It cannot perform P's part of the protocol for the remaining blocks which do correspond to a Hamiltonian cycle. In order to solve this problem, M modifies σ in a special way to obtain a new string σ'. Let B_l be any block which in σ corresponds to a Hamiltonian cycle. M replaces each *edge* entry in this block by a randomly constructed encoding of a *nonedge* entry. Thus B_l in σ' corresponds to a graph with no edges. For such a hidden graph S, M can simulate Protocol 2.4 even without knowing a cycle in G, because the entries that M does not open in S no longer contain a Hamiltonian cycle.

The simulation differs from a real execution only in one thing: All the unopened hidden bits in the simu-

lation represent *nonedges*, whereas in the real execution some of them represent *edges*. In the full paper we show that the ability to distinguish (in polynomial time) between the simulation and the real execution implies a nonuniform algorithm for inverting f on a nonnegligible fraction of the images, contradicting its onewayness. The proof is by the "hybrid argument" of Goldwasser and Micali [GM84]. ◇

2.4 Polynomial Time Provers

The protocol described in the previous subsection requires an exponential time prover in order to invert the oneway permutation and recover the hidden bits. But in cryptographic applications which rely on NIZK proofs, such as the signature scheme of [BG89] and the encryption scheme of [NY90], the prover is only polynomial time with an NP witness as auxiliary input. In order to adapt the protocol to this new setting, we need a stronger cryptographic assumption: The existence of families of trapdoor permutations. The random polynomial time prover selects a random member f_r of this family, sends it to the verifier, and keeps the trapdoor information to himself. Now he is in the position that he can invert f_r, and thus recover the hidden bits of σ with respect to f_r, whereas the verifier cannot.

Theorem 2.6 *Under the assumption that trapdoor permutations exist, there exist bounded noninteractive zero knowledge proofs of Hamiltonicity in which the prover is probabilistic polynomial time.*

Proof: Use the protocol of Subsection 2.3 with oneway permutations replaced by trapdoor permutations. Another minor change which has to be made is to increase the number of blocks by a factor of n, in order to retain the soundness property with respect to cheating provers who choose the "best" (rather than a random) member of the family of trapdoor permutations. Details omitted. ◇

3 Multiple NIZK Proofs Based on a Single Random String

3.1 Introduction

Using a bounded NIZK proof system the prover can prove one statement in zero knowledge. It is not known whether in general the zero knowledge property is preserved if the same common reference string σ is used to prove more than one statement. This is an obvious drawback when cryptographic applications are concerned.

Definition 3.7 *A noninteractive proof systems for the language L_R is* general zero knowledge *if there exists a random polynomial time simulator M such that for any polynomial sequence of instances $(x_1, w_1) \in R$, $(x_2, w_2) \in R$, $(x_3, w_3) \in R$, ... the two ensembles $(\sigma, x_1, P(x_1, w_1, \sigma), x_2, P(x_2, w_2, \sigma), ...)$ and $M(x_1, x_2, ...)$ are polynomially indistinguishable.*

Remark: Since the prover's algorithm in a general NIZK proof system is publicly available, polynomially many provers can share the same random reference string σ and prove polynomially many statements independently. This feature is not available in [BDMP89]'s construction of multiple NIZK proofs based on a single random reference string, since in their construction only the prover who proves the first statement knows how to construct NIZK proofs of subsequent statements.

In this section we show how to transform any bounded NIZK proof system for an NP complete language L_R into a general NIZK proof system for the same language L_R. We only consider NIZK proof systems in which the real prover need not be stronger than polynomial time. Our transformation does not apply to NIZK proof systems in which the prover is exponential time (such as that of Section 2.3 in which P inverts oneway permutations).

We now give a quick overview of our construction. It is based on the concept of *witness indistinguishability* [FS90], which informally means that it is intractable to distinguish which of two possible witnesses P is using in his proof for an NP statement. In [FS90] it is proved that any NIZK proof system in which the prover need not be stronger than polynomial time is also witness indistinguishable. Furthermore, the witness indistinguishability property is preserved even if polynomially many noninteractive witness indistinguishable proofs are given using the same common reference string (again, provided that the prover in each individual proof need not be stronger than polynomial time).

The main step in our construction of general NIZKs is to show that any sequence of noninteractive witness indistinguishable proofs is also zero knowledge. But this claim is not always true. To solve this problem we show how, under the assumption that oneway functions exist, one can modify any NIZK proof system for any NP complete language to a new noninteractive proof system for which witness indistinguishability always implies zero knowledge.

3.2 Noninteractive Witness Indistinguishability

In this subsection we recall the definitions and results related to noninteractive witness indistinguishability, as presented in [FS90].

Definition 3.8 *Noninteractive proof system (P,V) is* bounded witness indistinguishable *over R if for any large enough input x, any $w_1, w_2 \in w(x)$, and for a randomly chosen reference string σ, the ensembles which differ only in the witness that P is using, but not in x or σ, are indistinguishable. Formally:*

$\forall D\, \forall x \in L_R\ \forall w_1, w_2 \in w(x)$

$|\text{Prob}(D(x,\sigma,P(x,w_1,\sigma)) = 1)$

$-\text{Prob}(D(x,\sigma,P(x,w_2,\sigma)) = 1)| < \nu(n)$

The probability space is that of the random choices of σ together with P's random coin tosses.

Lemma 3.9 *Any bounded noninteractive zero knowledge proof system with polynomial time prover is also a bounded noninteractive witness indistinguishable proof system.*

The proof of this Theorem is given in [FS90] (Theorem 5.2).

Definition 3.10 *A noninteractive proof system is* general witness indistinguishable *over R if for any polynomial sequence of instances with respective pairs of witnesses, the two ensembles*
$(\sigma, x_1, P(x_1, w_1^1, \sigma), x_2, P(x_2, w_2^1, \sigma), \ldots)$ *and*
$(\sigma, x_1, P(x_1, w_1^2, \sigma), x_2, P(x_2, w_2^2, \sigma), \ldots)$ *are polynomially indistinguishable.*

Lemma 3.11 *Any bounded noninteractive witness indistinguishable proof system is also general witness indistinguishable.*

Proof: Assume that for infinitely many n there exist polynomial sets of inputs $\mathcal{X}(n)$ and sets of witnesses $\mathcal{W}^1(n)$ and $\mathcal{W}^2(n)$, such that the two ensembles $P(\mathcal{X}, \mathcal{W}^1, \sigma)$ and $P(\mathcal{X}, \mathcal{W}^2, \sigma)$ are polynomially distinguishable. By the "hybrid" argument of [GM84], there must be a "polynomial jump" somewhere in the execution: For any n, there exists k, such that if for all $x \in \mathcal{X}(n)$ with index less than k the prover uses witnesses from $\mathcal{W}^1$, and for all $x \in \mathcal{X}(n)$ with index greater than k the prover uses witnesses from $\mathcal{W}^2$, then the ensembles which differ only in the witness used for x_k are distinguishable by some distinguisher D. We use the nonuniformity of the distinguishers to derive a contradiction. The whole set of proofs can be simulated by a modified D', who has as auxiliary input the inputs and witnesses to all other proofs. We use here the fact that the prover is polynomial time, and that there are only polynomially many inputs of polynomial length. This random polynomial time D' can now distinguish between proofs for $x_{k(n)}$ in which the prover uses $w^1_{k(n)}$ and proofs in which the prover uses $w^2_{k(n)}$. This contradicts our assumption that the original protocol was witness indistinguishable. ◇

3.3 The Transformation

We assume the existence of cryptographically secure pseudo-random bit generators [BM84], which extend n-bits random seeds to $2n$-bits pseudo-random strings, polynomially indistinguishable from strings of truly random $2n$ bits. The existence of pseudo-random generators follows from the assumption that oneway functions exist ([ILL89], [H90]).

Let (P,V) be any bounded NIZK proof system with polynomial time prover for the NP complete language L_R. We construct $(\hat{P}, \hat{V})$, a general NIZK proof system for L_R, under the sole assumption that oneway functions exist.

We divide the common random string σ into two segments: The first $2n$ bits which we call the *reference statement* y, and the rest of the common random string, which will be used as a reference string σ' for the bounded protocol (P,V). The reference statement is interpreted as "y is a pseudo-random string". This is an NP statement, whose witness is the n bit seed which generates y (if such a seed exists). On input $(x,w) \in R$, prover $\hat{P}$ constructs a new NP statement $x\#y$. A witness to this statement is any string w' which either satisfies $(x,w') \in R$, or is a seed for y. $\hat{P}$ reduces $x\#y$ to an instance X of the NP complete language L_R, using a publicly known witness preserving reduction (known reductions have these properties). $\hat{P}$ reduces the witness w that he has for $x \in L_R$ to a witness for $X \in L_R$, and uses the system (P,V) and the reference string σ' to prove that $X \in L_R$.

Theorem 3.12 *Under the assumptions that (P,V) is a bounded NIZK proof system and that oneway functions exist, the above transformed scheme is a general noninteractive zero knowledge proof system for L_R.*

Proof: We first give an intuitive introduction to the full proof. The completeness, soundness and zero knowledge properties of $(\hat{P}, \hat{V})$ are based on the corresponding properties of (P,V). The completeness property follows from the fact that from a witness to x, prover $\hat{P}$ can derive a witness to X, and thus execute the bounded NIZK proof system (P,V). The soundness property follows from the fact that y is chosen as a truly random (rather than pseudo-random)

string. Thus, for almost all possible choices of y, $X \in L_R$ (allowing $\hat{P}$ to execute (P,V)) only if $x \in L_R$. The zero knowledge property requires more subtle analysis. The simulation of $(\hat{P}, \hat{V})$ is done by replacing the reference statement y by a pseudo-random string y'. This replacement is indistinguishable to polynomial time observers. Now any statement x with witness w is transformed into a statement X which also has the seed of y' as its witness. The simulator uses this seed instead of w in order to prove X. The concept of witness indistinguishability can now be used to show that this change of witnesses in the proof of X is indistinguishable to polynomial time observers, even if it is done polynomially many times. We now give a more detailed proof.

Completeness: The reduction to L_R preserves witnesses. Thus $\hat{P}$, who knows a witness for x, can also compute in polynomial time a witness for X and use it in order to perform the protocol. The reduction is also publicly known, and so $\hat{V}$ can check that it was followed correctly. The completeness property then follows from the completeness property of (P,V).

Soundness: From the soundness property of (P,V) it follows that either $x \in L_R$, or y is pseudorandom. But simple counting shows that the probability that the random string y of length $2n$ is pseudorandom (i.e., has a seed of length n) is at most 2^{-n}, and thus with overwhelming probability indeed $x \in L_R$.

The completeness and soundness property trivially continue to hold even if polynomially many statements are proved.

Zero knowledge: Assume $\hat{P}$ proves membership in L_R of polynomially many inputs $x_1, x_2, \ldots$ (using the associated witnesses $w_1, w_2, \ldots$). We construct a simulator M which creates an ensemble indistinguishable from the ensemble that $\hat{P}$ produces.

M randomly selects an n bit seed s, and pseudorandomly generates the $2n$-bits string y', to be used as the reference statement (instead of a random y used in reality). M generates a truly random reference string σ'. For each $j \geq 1$, M reduces $x_j \# y'$ to an instance X_j of L_R, and derives from s a witness w' for this instance. Then M uses the proof system (P,V) and the reference string σ' to prove that $X \in L_R$, by using his knowledge of the seed s rather than knowledge of the witnesses w_j to the statements x_j.

In order to prove that M's simulation is indistinguishable from $\hat{P}$'s proofs, we construct a hybrid $\bar{M}$, which constructs y' pseudo-randomly as M does, but gives the proofs using $w_1, w_2, \ldots$ as $\hat{P}$ does.

Lemma 3.13 *$\bar{M}$'s output is indistinguishable from $\hat{P}$'s output.*

Proof: Otherwise, $\hat{P}((x_1, w_1), (x_2, w_2), \ldots)$ forms a nonuniform polynomial time statistical test for the pseudo-randomness of y, which is a contradiction. $\diamond$

We now prove that the outputs of M and $\bar{M}$ are indistinguishable. Protocol (P,V) is zero knowledge. By Lemma 3.9 it is witness indistinguishable. By Lemma 3.11, even polynomially many executions of (P,V) on the same reference string are witness indistinguishable. $\bar{M}$ and M differ only by the witnesses that they are using, and so by the witness indistinguishability property of the protocols, the ensembles that they create are indistinguishable. $\diamond$

Remark: The first step in proving that our general construction is zero knowledge was to replace the zero knowledge property of the bounded proof system by witness indistinguishability. Our task would have been easier had we started directly with a witness indistinguishable bounded noninteractive proof system rather than with a zero knowledge one. Consequently, in order to construct general NIZK proof systems, it is sufficient to construct a bounded NIWI proof system. This may be an easier task than constructing a bounded NIZK proof system, because proving the WI property of a protocol does not involve the design of a simulator M.

Remark: In bounded NIZK proof systems, the length of the common random string bounds the size of the NP statement that can be proved. [BDMP89] show that with general NIZK proof systems statements of unbounded size can be proven. Their technique can be adapted to our case as well (details omitted).

4 Security Against Adaptive Attacks

4.1 Definitions

Noninteractive zero knowledge proof systems are useful design primitives in the construction of cryptographic schemes, such as signature schemes [BG89] and encryption schemes [NY90]. It is often required that the cryptographic scheme will be robust against attacks of adaptive nature, which are the strongest types of attack. For example, a standard security requirement of signature schemes is that even after requesting signatures of polynomially many messages of his choice, the adversary is not able to forge a signature to any new message. In order to treat adaptive attacks we extend the security requirements of NIZK proof systems.

In a typical adaptive scenario, a polynomial time adversary A repeatedly selects statements and observes their noninteractive proofs. His goal is to come

up with a statement x on which one of the three basic properties of NIZK proof systems is violated: Either x is true but P cannot produce a noninteractive proof for it (violating the *completeness* condition), or x is false but a noninteractive "proof" to the contrary can be forged (violating the *soundness* condition), or x is true and the adversary can extract useful information from the noninteractive proof that P gives (violating the *zero knowledge* property).

Definition 4.14 *A* noninteractive adaptive proof system *for an NP language L_R characterized by a relation R is a pair of probabilistic polynomial time algorithms (P, V) satisfying:*

1. Adaptive completeness: *For any nonuniform polynomial time adversary A which on input σ generates $(x, w) \in R$,*
   $\text{Prob}(V(x, \sigma, P(x, w, \sigma)) \text{ accepts}) > 1 - \nu(n)$
2. Adaptive soundness: *For any nonuniform polynomial time adversary A which on input σ generates $x \notin L_R$,*
   $\text{Prob}(V(x, \sigma, A(x)) \text{ accepts}) < \nu(n)$

The probabilities are taken over the choices of σ, and over the coin tosses of A, P and V.

Proposition 4.15 *Any noninteractive adaptive proof system (Definition 4.14) is a noninteractive proof system (Definition 1.2).*

Proof: Follows directly from the nonuniformity of the adversary in Definition 4.14. ◇

The converse of the above theorem is not necessarily true. There is no a-priori reason why the usual completeness and soundness conditions should imply the adaptive ones. But it turns out that more often then not, they do. Consider for example the NIZK proof system of Section 2. It has *perfect completeness* (i.e., the verifier *always* accepts P's proofs of true statements), and thus it also has adaptive completeness. Likewise, for almost all possible choices of σ, the proof system has *perfect soundness* (i.e., no false statement can be proven), and thus it also has adaptive soundness.

We define the concept of *adaptive zero knowledge* by a test similar to [GGM86]'s test for pseudo-random functions. We call this test *adaptive indistinguishability*, or the AI test (partially because of its origins as Turing's test for artificial intelligence). In our AI test, an adversary A is confronted with a blackbox B. His goal is to determine whether B contains a real prover P, or whether it contains a simulator M. A first requests the random reference string σ from B. If P is inside the box, it replies with a truly random string. If M is inside the box, it replies with a string of its choice. Now, possibly based on σ, A generates a pair $(x, w) \in R$ and sends it to B. If P is inside the box, it produces a noninteractive proof $P(x, w, \sigma)$. If M is inside the box, w is magically filtered away, and M must simulate a noninteractive proof for x. This procedure of adaptively choosing theorems and receiving noninteractive proofs for them is repeated polynomially many times until A is ready to pass a decision: '0' or '1'. (M, P) are said to pass the AI test if the probabilities that A outputs '1' when M is inside the box and when P is inside the box are equal up to negligible additive terms.

Definition 4.16 *A noninteractive adaptive proof system (P, V) is* adaptive zero knowledge *if there exists a random polynomial time simulator M such that (M, P) pass the AI test for any nonuniform polynomial time adversary A.*

Proposition 4.17 *Any noninteractive proof system which is adaptive zero knowledge (Definition 4.16) is also general zero knowledge (Definition 3.7).*

Proof: Follows directly from the nonuniformity of the adversary in Definition 4.16. ◇

A somewhat weaker condition than adaptive zero knowledge is *single statement adaptive zero knowledge*. For such proof systems, the adversary of the AI test is allowed to request only one noninteractive proof from B before passing his judgement as to what is inside the box.

Proposition 4.18 *Any noninteractive proof system which is single statement adaptive zero knowledge is also bounded zero knowledge (Definition 1.3).*

Proof: Follows directly from the nonuniformity of the adversary. ◇

Remark: The converse of the above theorem is probably not true. See for example the NIZK in [BFM88] proving that x is a product of two primes. If x is chosen adaptively after the simulator generates σ, it is not clear how M can complete the simulation of the noninteractive proof.

4.2 Robustness of Our Protocols

Theorem 4.19 *The NIZK of Section 2.4 is single statement adaptive zero knowledge.*

Proof (sketch): Consider the simulator M described in Section 2. This simulator generates a reference string σ' independently of the common input x. This σ' can be used to simulate a noninteractive

proof for any input x. Consequently, the simulation process is uneffected even if the input statement is chosen adaptively after the reference string is chosen. It remains to show that the polynomial time adversary A cannot distinguish between M and truthful P on the basis of a proof of a single statement. In the full paper we prove that the contrary implies that A can also be used as an algorithm which inverts instances of trapdoor permutations with nonnegligible probability, contradicting their onewayness. ◇

Theorem 4.20 *The transformation of Section 3.3 transforms noninteractive single statement adaptive zero knowledge proof systems into noninteractive adaptive zero knowledge proof systems.*

Proof (sketch): The proof of the completeness and soundness conditions is similar to their proof in Theorem 3.12, and is omitted. The proof of the adaptive zero knowledge property is based on the following notion of *adaptive witness indistinguishability.*

Definition 4.21 *A noninteractive adaptive proof system (P,V) is* adaptive witness indistinguishable *if (P_1,P_2) pass the AI test for witness indistinguishability. In this test B generates a random reference string σ, the nonuniform polynomial time adversary A adaptively generates a sequence of triplets (x_i,w_i^1,w_i^2), where $(x_i,w_i^1)\in L_R$ and $(x_i,w_i^2)\in L_R$, and P_1 uses only the first of the given witnesses for x to generate the noninteractive proofs $P(x_i,w_i^1,\sigma)$, whereas P_2 uses the second witness.*

Single statement witness indistinguishability is the parallel of single statement zero knowledge.

Lemma 4.22 *Any noninteractive proof system which is single statement adaptive zero knowledge is also single statement adaptive witness indistinguishable.*

Proof (sketch): Assume (P,V) is not single statement adaptive witness indistinguishable. Then there exists an adversary A which when given σ generates a triplet (x,w_1,w_2) and distinguishes between real provers of type P_1 which use the first witness to produce a noninteractive proof $P(x,w_1,\sigma)$, and real provers of type P_2 which use the second witness to produce $P(x,w_2,\sigma)$.

In order to prove that (P,V) cannot be zero knowledge, we define two types of adversaries: A_1 and A_2. When confronted with a blackbox B that could be either the real prover P or a simulator M, adversary A_i, for $i=1$ *or* 2, generates (x,w_1,w_2) as A does, feeds B with (x,w_i), and outputs its final decision ('0' or '1') as A does. When P is inside the box, then by our assumption the probability that A_1 outputs '1' differs from the probability that A_2 outputs '1' by at least n^{-k}, for some positive constant k. When M is inside the box, the two adversaries output '1' with the same probability. Consequently, (P,M) fail the AI test with respect to at least one of the two adversaries. ◇

The following lemma shows that adaptive witness indistinguishability is preserved under repeated applications of the noninteractive proof system with the same random reference string.

Lemma 4.23 *Any noninteractive proof system which is single statement adaptive witness indistinguishable is also adaptive witness indistinguishable.*

Proof (sketch): Assume that there exists an adversary A which adaptively generates triplets (x_i,w_i^1,w_i^2) (for $i\geq 1$) and can distinguish between P_1 and P_2. By the "hybrid" argument of [GM84], there must be a "polynomial jump" somewhere in the execution: There exists k, such that if for $i<k$ the adversary uses the first of the two generated witnesses of each instance to produce $P(x_i,w_i^1,\sigma)$ by himself, then generates (x_k,w_k^1,w_k^2) and gives it to the blackbox, and finally (for $i>k$) uses the second of the two generated witnesses of each instance to produce $P(x_i,w_i^2,\sigma)$ by himself, then A can distinguish between the case that P_1 is inside the box and the case that P_2 is inside the box. This contradicts our assumption that the original protocol was single statement adaptive witness indistinguishable. ◇

Having established Lemmas 4.22 and 4.23, which are the adaptive parallels of Lemmas 3.9 and 3.11 respectively, Theorem 4.20 is proved in exactly the same way as Theorem 3.12, which is its nonadaptive counterpart. ◇

4.3 Uniform Cryptographic Assumptions

Throughout this paper we treat the concept of NIZK under nonuniform cryptographic assumptions (e.g., we assumed that there exist pseudo-random generators secure with respect to nonuniform polynomial time adversaries). Goldreich [G89] shows how zero knowledge interactive proofs can be dealt with in a uniform complexity setting. The same can be done for noninteractive zero knowledge proofs. For this purpose, it is necessary to modify the definitions of NIZK proof systems. More specificly, the nonuniform polynomial time adversary of Definitions 4.14 and 4.16 should be replaced by a random polynomial time adversary. With this change of definitions, it is

possible to derive all the results of this paper under uniform complexity assumptions, though the proofs of correctness become somewhat more complicated. Details appear in the full paper.

Acknowledgements

We thank Moni Naor for helpful discussions and Oded Goldreich for his useful remarks on an earlier manuscript.

References

[BG89] M. Bellare, S. Goldwasser, "New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero Knowledge Proofs", *Proc. of Crypto89, pp. 194-211.*

[Blum86] M. Blum, "How to Prove a Theorem So No One Else Can Claim It" *Proc. of the International Congress of Mathematicians, Berkeley, California, USA, 1986, pp. 1444-1451.*

[BDMP89] M. Blum, A. De Santis, S. Micali, G. Persiano, "Non-Interactive Zero Knowledge" *MIT/LCS/TM-430.*

[BFM88] M. Blum, P. Feldman, S. Micali, "Non-interactive Zero Knowledge and its Applications" *Proc. of 20th STOC 1988, pp. 103-112.*

[BM84] M. Blum, S. Micali, "How to Generate cryptographically Strong Sequences of Pseudo-Random Bits" *SIAM Jour. on Computing, Vol. 13, 1984, pp. 850-864.*

[DMP87] A. De Santis, S. Micali, G. Persiano, "Non-Interactive Zero-Knowledge Proof Systems" *Proc of CRYPTO-87, pp. 52-72.*

[DMP88] A. De Santis, S. Micali, G. Persiano, "Non-Interactive Zero-Knowledge with Preprocessing" *Proc of CRYPTO-88, pp. 269-283.*

[DY] A. De Santis, M. Yung, "Metaproofs (and their Cryptographic Applications)" *manuscript, 1990.*

[FS90] U. Feige, A. Shamir, "Witness Indistinguishable and Witness Hiding Protocols", *Proc. of 22nd STOC, 1990, pp. 416-426.*

[G89] O. Goldreich, "A Uniform-Complexity Treatment of encryption and Zero-Knowledge" *TR-568, Computer Science Dept., Technion, Haifa, Israel, 1989.*

[GGM86] O. Goldreich, S. Goldwasser, S. Micali, "How to Construct Random Functions" *Jour. of ACM, Vol. 33, No. 4, 1986, pp. 792-807.*

[GL89] O. Goldreich, L. Levin, "A Hard-Core Predicate for all Oneway Functions" *Proc. 21st STOC 1989, pp. 25-32.*

[GMW86] O. Goldreich, S. Micali, A. Wigderson, "Proofs that Yield Nothing But Their Validity and a Methodology of Cryptographic Protocol Design", *Proc. 27th FOCS, 1986, pp. 174-187.*

[GM84] S. Goldwasser, S. Micali, "Probabilistic Encryption" *JCSS, Vol. 28, No. 2, 1984, pp. 270-299.*

[GMR85] S. Goldwasser, S. Micali, C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems", *SIAM J. Comput. Vol. 18, No. 1, pp. 186-208, February 1989. Preliminary version in Proc. of 17th STOC, 1985, pp. 291-304.*

[H90] J. Hastad, "Pseudo-Random Generators under Uniform Assumptions" 22nd *STOC, 1990, pp. 395-404.*

[ILL89] R. Impagliazzo, L. Levin, M. Luby, "Pseudorandom Generation from Oneway Functions" 21st *STOC, pp. 12-24, 1989.*

[KMO89] J. Kilian, S. Micali, R. Ostrovsky, "Minimum Resource Zero-Knowledge Proofs" *Proc. of FOCS89.*

[LS90] D. Lapidot, A. Shamir, "Publicly Verifiable Non-Interactive Zero-Knowledge Proofs" *Proc. of Crypto 90.*

[Naor89] M. Naor, "Bit Commitment Using Pseudorandomness" *Proc. of Crypto89, pp. 128-137.*

[NY90] M. Naor, M. Yung, "Public-key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks" *Proc. of STOC90, 427-437.*