Witness Indistinguishable and Witness Hiding Protocols

Uriel Feige, Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract

A two party protocol in which party A uses one of several secret witnesses to an NP assertion is *witness indistinguishable* if party B cannot tell which witness A is actually using. The protocol is *witness hiding* if by the end of the protocol B cannot compute any new witness which he did not know before the protocol began. Witness hiding is a natural security requirement, and can replace zero knowledge in many cryptographic protocols.

We prove two central results: 1. Unlike zero knowledge protocols, witness indistinguishablity is preserved under arbitrary composition of protocols, including parallel execution. 2. If a statement has at least two independent witnesses, then any witness indistinguishable protocol for this statement is also witness hiding.

Using these results, we show how to overcome some of the difficulties associated with cryptographic schemes based on zero knowledge protocols. In particular, we show how to parallelize identification protocols without loss of security, how to construct bounded round zero knowledge arguments for any NP statement under the sole assumption that oneway functions exist, and how to use the Bellare-Goldwasser signature scheme to sign polynomially many messages in a completely memoryless way.

1 Introduction

This paper introduces the concepts of *witness indistinguishable* (WI) and witness hiding (WH) protocols, develops the basic theory and demonstrates several cryptographic applications of these concepts. We deal with two party protocols, in which both P (prover) and V (verifier) are polynomial time. P and V see a common input x, and P has a secret auxiliary input: a *witness* w from the witness set $w(x)$. P's purpose is to perform a computational task that would be difficult to perform had P not known w. Typical examples are to give an interactive proof that he "knows" a witness to the NP statement x (in this case w may be the NP witness), or to digitally sign messages which can later be checked by the public key x (in this case w is P's private key). Informally, these protocols are *witness indistinguishable* if V cannot distinguish between two executions of the protocol which differ in the specific witness P is using. For example, let x be a Hamiltonian graph with several Hamiltonian cycles. A proof of Hamiltonicity is WI if V's view is the same no matter which cycle w in x party P knows and uses.

In order to use the WI property for cryptographic security, the protocol must be nontrivial (any protocol is trivially WI if the witness set $w(x)$ contains only one witness). We are interested in protocols in which V cannot learn any witness from the protocol, which we call *witness hiding* (WH) protocols. An important part of the paper is devoted to showing that if a protocol is WI, and if $w(x)$ contains at least two independent witnesses, then the protocol must be WH. The WH property is a natural property which is sufficient to guarantee overall security of many cryptographic schemes (nontransitivity of proofs of knowledge, unforgeable proofs of identity etc.).

It is natural to compare the concept of WH to that of *zero knowledge* (ZK [15]). ZK guarantees that no information whatsoever leaks during the execution of

a protocol. WH only guarantees that the prover's witness does not leak, and says nothing about other information. Thus, in some cryptographic applications, one would prefer to use ZK protocols instead of WH protocols. Unfortunately, ZK is not preserved under general composition of protocols, and this limits its applicability. On the other hand, we show that the WI property is preserved under arbitrary composition of protocols (including parallel composition). Thus, unlike the case for ZK protocols, one may plug a WI protocol into any cryptographic scheme and be sure that the protocol retains its qualities, including the property of being witness hiding. The applications which we suggest for WI and WH protocols are applications where compositionality is required.

Outline of the paper and main results: Section 2 contains background material (notation, basic definitions). In section 3 we define the concept of witness indistinguishability and prove that it is preserved under polynomial composition of protocols. We contrast this with a particular zero knowledge protocol which discloses P's witness when executed twice in parallel, demonstrating that ZK is not closed under general composition. The section ends with a simple methodology for constructing WI protocols. Section 4 introduces the concept of witness hiding. It is devoted to proving that any WI protocol for statements which have two independent witnesses is also WH. The proof also gives a methodology for constructing inputs which have two independent witnesses. Section 5 offers cryptographic applications of the new concepts: Constructions of secure identification schemes, unforgeable signature schemes, and bounded rounds zero knowledge arguments for any NP statement.

Related work: Some familiarity with the zero knowledge concept ([15], [13]) will help in understanding the new concepts of witness indistinguishability and witness hiding. The concept of *transferable information* [9] contains the seeds to many of the ideas presented here. The fact that the zero knowledge property is not preserved under parallel composition was conjectured by many and recently proved in [12]. Our proof of this fact is of independent interest, since the proof in [12] assumes the provers are computationally unbounded, and uses it in an essential way. In contrast, we prove that zero knowledge is not preserved under parallel composition even if the provers are only polynomial time. The application of WH subprotocols to the construction of bounded round zero knowledge arguments for any NP statement is described in detail in [8]. Other bounded round zero knowledge arguments for any NP statement are known ([5], [11]), but our protocols rely on a weaker cryptographic assumption (the existence of one way functions) and require fewer rounds. The signature scheme we construct is an impovement of the one presented in [6], and allows any user to sign any polynomial number of messages in a completely "memoryless" fashion.

2 Notation and Definitions

For a discussion of the following definitions, see [15] (interactive proofs and zero knowledge),[4], [20] and [9] (proofs of knowledge).

Our model of computation is the probabilistic polynomial time interactive Turing machine (both for the prover P and for the verifier V). The common input is denoted by x, and its length is denoted by $|x| = n$. Each machine has an auxiliary input tape. P's auxiliary input is denoted by w. V's auxiliary input is denoted by y. $\nu(n)$ denotes any function vanishing faster than the inverse of any polynomial. Formally:

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

Negligible probability is probability behaving as $\nu(n)$. *Overwhelming* probability is probability behaving as $1 - \nu(n)$.

$A(x)$ denotes the output of a probabilistic algorithm A on input x. This is a random variable. $V_P(x)$ denotes V's output after interaction with P on common input x. $M(x; A)$ (where A may be either P or V) denotes algorithm M's output on input x, where M may use algorithm A as a (blackbox) subroutine. Each call M makes to A is counted as one computation step for M.

Definition 2.1: Let R be a relation $\{(x, w)\}$ testable in polynomial time, where $|x| = |w|$ (this restriction can be met by standard padding techniques for all relations of interest). For any x, its *witness set* $w(x)$ is the set of w such that $(x, w) \in R$. $\diamond$

Definition 2.2: An *interactive proof of knowledge* system for relation R is a pair of algorithms (P, V) satisfying:

1. *Completeness:* $\forall (x, w) \in R$
   $Prob(V_{P(x,w)}(x) accepts) > 1 - \nu(n)$
2. *Soundness:* $\exists M \; \forall P' \; \forall x \; \forall w'$
   $Prob(V_{P'(x,w')}(x) accepts) <$
   $Prob(M(x, w'; P') \in w(x)) + \nu(n)$

The probability is taken over the coin tosses of V, P' and M. The knowledge extractor M runs in expected polynomial time, and uses P' as a blackbox. $\diamond$

Remark: If $w(x)$ is empty, this definition implies that the probability that V accepts is negligible. $\diamond$

Definition 2.3: Proof system (P, V) is *zero knowledge* (ZK) over R if there exists a simulator M which

runs in expected polynomial time, such that for any probabilistic polynomial time V', for any $(x,w) \in R$, and any auxiliary input y to V', the two ensembles $V'_{P(x,w)}(x,y)$ and $M(x,y;V')$ are polynomially indistinguishable. M is allowed to use V' as a subroutine. ◇

3 Witness indistinguishability

Informally, a protocol is witness indistinguishable if the verifier cannot tell which witness the prover is using (even if the verifier knows all witnesses to the statement being proved).

Definition 3.1: Proof system (P,V) is *witness indistinguishable* (WI) over R if for any V', any large enough input x, any $w_1, w_2 \in w(x)$, and for any auxiliary input y for V', the ensembles, $V'_{P(x,w_1)}(x,y)$ and $V'_{P(x,w_2)}(x,y)$, generated as V's view of the protocol, are indistinguishable.◇

Remark: Unlike definition 2.3 (for ZK), the definition for WI involves no simulator M.

In this section we show that WI is preserved under general composition of protocols, while ZK is not.

Imagine a cryptographic community with several types of protocols: Identification protocols, key exchange protocols, etc. Each type of protocol may be executed many times, with different inputs and different participating parties. Each party may take part in several protocols, and execute these protocols with any sort of interleaving between their steps. Though it may not be the intention of the designer of such a system to compose together several protocols, this situation may be created by a (cheating) party which uses information from the execution of some protocols in which it is participating in order to compute its responses in other protocols. In order to discuss the effect of these arbitrary compositions on the WI property, we allow all the parameters of the system to grow and consider asymptotics. Thus, if n is a parameter denoting size, then the number of parties, their running time, the sizes of their descriptions, and the sizes of individual inputs to each protocol are all bounded by some polynomial in n. On the other hand, the types of protocols which can be run in such a system are fixed in advance, and their number does not grow with n.

Definition 3.2: For some constant t, let R^j (for $1 \leq j \leq t$) be relations testable in polynomial time, and let (P^j, V^j) be respective proof systems for these relations. The *general composition* of protocols (P_i, V_i), each one of which is from one of the t types mentioned above, is their concurrent execution, with any sort of interleaving between the elementary steps of different protocols. For convenience we assume that the inputs x_i to the protocols are all of the same size n. The provers and verifiers in the different protocols need not be pairwise distinct, and the inputs x_i and the relations R^i need not be pairwise distinct. The composition is *polynomial* if there is one polynomial in n bounding the number of participants, the sizes of their descriptions, and their running times. ◇

We single out two special cases of the general composition: *Sequential composition*, in which the protocols are executed one after the other, and *parallel composition*, in which all protocols are of the same type and run on the same input, and for each j, steps j in all the protocols are executed at the same time.

Definition 3.3: A polynomial composition of protocols is *witness indistinguishable*, if for any subset of provers $\mathcal{P} = (P_1, P_2, ..., P_k)$ which follow their protocols faithfully, and any two sets of respective witnesses $\mathcal{W}^1 = (w_1^1, w_2^1, ...w_k^1)$ and $\mathcal{W}^2 = (w_1^2, w_2^2, ...w_k^2)$, it is indistinguishable to the coalition of all the other provers and verifiers whether $\mathcal{P}$ are using $\mathcal{W}^1$ or $\mathcal{W}^2$ (for large enough n, where k may be a function of n).◇

Theorem 3.1: WI is preserved under polynomial composition of protocols.

Proof (sketch): Consider polynomially many protocols carried out concurrently (sequentially, in parallel, or with interleaved steps). Assume that for infinitely many n, $\mathcal{P}(n)$ are subsets of the provers who carry out their WI protocols faithfully and for them WI is not preserved. That is, there exist sets of verifiers $\mathcal{V}(n)$, auxiliary inputs $\mathcal{Y}(n)$ to $\mathcal{V}(n)$, and sets of witnesses $\mathcal{W}^1(n)$ and $\mathcal{W}^2(n)$, such that the two ensembles $\mathcal{V}_{\mathcal{P}(\mathcal{W}^1)}(\mathcal{Y}(n))$ and $\mathcal{V}_{\mathcal{P}(\mathcal{W}^2)}(\mathcal{Y}(n))$ are polynomially distinguishable. By the "hybrid" argument of [14], there must be a "polynomial jump" somewhere in the execution: For any n, there exists k, such that if all $P \in \mathcal{P}(n)$ with index less than k use witnesses from $\mathcal{W}^1$, and all $P \in \mathcal{P}(n)$ with index greater than k use witnesses from $\mathcal{W}^2$, the ensembles which differ only in the witness P_k is using are distinguishable by $\mathcal{V}$. Now we use the auxiliary input of the verifier to derive a contradiction. The whole set of protocols which are taking place concurrently can be simulated by a modified V', who has as auxiliary input the algorithms and auxiliary inputs of all other participants (including $\mathcal{Y}(n)$, $\mathcal{W}^1(n)$ and $\mathcal{W}^2(n)$). We use here the fact that the composition is polynomial: there are only polynomially many participants, and each one of them is polynomial time (including the provers). This random polynomial time V' can now distinguish between truthful $P_{k(n)}$ using $w^1_{k(n)}$ or $w^2_{k(n)}$. Since there are only finitely many (t) types of protocols (by Def-

inition 3.2), then the WI property is violated on infinitely many inputs for at least one of these types of protocols. This contradicts our assumption that the original protocol was witness indistinguishable. ◇

We now address the composition of zero knowledge protocols. We demonstrate that P's secret witness may be disclosed if a ZK protocol is composed twice with itself.

Theorem 3.2: There exists a zero knowledge proof of knowledge system ($\bar{P}$,$\bar{V}$) for the discrete log, which when executed twice in parallel discloses the discrete log of the input.

Proof(sketch): Let (P, V) be any zero knowledge proof of knowledge system for the discrete log problem (e.g. see [20]). We construct ($\bar{P}$,$\bar{V}$) directly from (P, V).

1. On input (p, g, x), $\bar{V}$ tries to randomly guess w, the unique discrete log of x, satisfying $g^w = x \bmod p$. If $\bar{V}$ succeeds (with negligible probability), he sends 1. Otherwise he sends 0.

2. If $\bar{V}$ sent 1 in move 1, he now proves to $\bar{P}$ in zero knowledge that he knows w, using the protocol (P, V) with reversed roles. If $\bar{P}$ is convinced by $\bar{V}$'s proof (this is expected to happen with overwhelming probability with truthful $\bar{P}$ and $\bar{V}$), he sends w to $\bar{V}$, showing that he too knows w, and $\bar{V}$ accepts. If $\bar{P}$ is not convinced by $\bar{V}$'s proof, $\bar{P}$ stops and $\bar{V}$ rejects.

3. If $\bar{V}$ sent 0 in move 1, $\bar{P}$ proves his knowledge of w using the standard proof system (P, V).

The protocol ($\bar{P}$,$\bar{V}$) is a complete and sound (perfect) zero knowledge proof of knowledge.

Consider now two executions, $(\bar{P}_1, \bar{V})$ and $(\bar{P}_2, \bar{V})$ in parallel. A cheating verifier V can always extract w from $\bar{P}_1$ and $\bar{P}_2$ using the following strategy: In move 1, V sends 0 to $\bar{P}_1$ and 1 to $\bar{P}_2$. Now V has to execute the protocol (P, V) twice: Once as a verifier talking to the prover $\bar{P}_1$, and once as a prover talking to the verifier $\bar{P}_2$. This he does by serving as an intermediary between $\bar{P}_1$ and $\bar{P}_2$, sending $\bar{P}_1$'s messages to $\bar{P}_2$, and $\bar{P}_2$'s messages to $\bar{P}_1$. Now $\bar{P}_2$ willfully sends w to V. ◇

Remark 1: Assuming the intractability of the discrete log, Theorem 3.2 proves that zero knowledge is not preserved under parallel composition.

Remark 2: We emphasize the importance of the fact that x has a *unique* witness w. Otherwise a single execution of the protocol ($\bar{P}$,$\bar{V}$) would not be zero knowledge, as it might reveal which of the witnesses for x $\bar{P}$ is using. This fact cannot be deduced by a simulator M just by observing x and $\bar{V}$.

Remark 3: Theorem 3.2 generalizes to any other relation $R = \{(x, w)\}$ which has a zero knowledge interactive proof of knowledge system, provided each instance has a unique witness.

Remark 4: Our result should not be confused with Theorem 7 in [12], which states that "Computational Zero-Knowledge is not closed under parallel composition". They state their Theorem in the model where the prover is infinitely powerful, and they use this in an essential way in the proof. Furthermore, they use in an essential way the fact that the protocols are only computational zero knowledge (and not perfect zero knowledge). We do not make any of these restrictions. On the other hand, our result relies on intractability assumptions, while [12] does not use unproven assumptions.

The next Theorem shows a simple relation between ZK and WI protocols.

Theorem 3.3: Let (P, V) be any ZK protocol. Then the protocol is WI.

Proof (sketch): The proof follows from the transitivity of the indistinguishability relation. For input x, assume distinguisher D has probability p of outputting 1 on V's view of P's proof, when P is using w_1. By the zero knowledge property, D has the same probability p (up to negligible additive terms) of outputting 1 on the simulated view created by M. But the view M creates is independent of the witness P is using, since M is not given such a witness. Thus D has probability p of outputting 1 on V's view even if P is using w_2. ◇

The above Theorems establish a methodology for constructing WI protocols. Take the basic step of a ZKIP. By Theorem 3.3 it is also WI. Iterate the basic step n times in parallel. This is probably not zero knowledge [12], but by Theorem 3.1, it is WI. In particular, we get:

Corollary 3.4: Under the assumption that oneway functions exist, any NP language has a constant round WI proof system.

4 Witness hiding

The concept of Witness Hiding (WH - to be defined shortly) is a possible alternative to zero knowledge. It is a weaker requirement than zero knowledge, but in many cases, it still satisfies the security demands of cryptographic protocols. Informally, a protocol (P, V) is WH if participating in the protocol does not help V to compute any new witnesses to the input which he did not know at the beginning of the protocol. This is a natural security requirement of cryptographic protocols. In order to prove the WH

property, one must show that if V' can compute a witness to the input after participating in the interactive proof, then he had this capability in him even before the protocol began. The definition of WH involves a probability distribution over the inputs. For this end, we borrow (and slightly modify) terminology from [1].

Definition 4.1: G is a *generator* for relation R if on input 1^n it produces instances $(x, w) \in R$ of length n. G is an *invulnerable generator* if for any polynomial time nonuniform *cracking* algorithm C, $Prob((x, C(x)) \in R) < \nu(n)$, where $x = G(1^n)$. The probability is taken over the coin tosses of G and C. $\diamond$.

Definition 4.2: Let (P, V) be a proof of knowledge system for relation R, and let G be a generator for this relation. (P, V) is *witness hiding* (WH) on (R, G) if there exists a witness extractor M which runs in expected polynomial time, such that for any nonuniform polynomial time V'

$Prob(V'_{P(x,w)}(x) \in w(x)) <$

$Prob(M(x; V', G) \in w(x)) + \nu(n)$

where $x = G(1^n)$. The probability is taken over the distribution of the inputs and witness, as well as the random tosses of P and M. The witness extractor is allowed to use V' and G as blackboxes. $\diamond$

There are two main differences between WH and zero knowledge:

1. The distribution on the inputs enters the definition (through G). There might be infinitely many inputs on which P willingly discloses his witness, but the protocol may still be WH if the probability of G picking such an input is negligible. This distribution on the inputs implies that V' must have the same auxiliary input for any common input of size n, unlike the case of ZK protocols, where V's auxiliary input may depend upon x.

2. The definition only guarantees that "whole" witnesses are not disclosed. Partial information may leak. In particular, the communication tape generated by $V'_{P(x,w)}(x)$ may not be simulatable in random polynomial time, and thus may serve as evidence that the protocol took place. In some cases this is an advantage. For example: Digital signatures cannot be zero knowledge (otherwise they are forgeable) and thus zero knowledge is an inadequate framework for defining their security. On the other hand, digital signatures can be witness hiding, hiding the auxiliary information which allows the true signer to sign messages.

What we need now is to establish a connection between WI and WH. We cannot prove that any WI protocol is also WH, but we can specify simple conditions under which WI implies WH. These conditions involve the particular method by which input instances are generated. A protocol may be trivially WI if the relation R is such that every input has only one possible witness. In this case WI cannot imply anything. Furthermore, Theorem 3.2 demonstrates that in this case one should not trust even ZK protocols in nonsequential compositions. But if each input has at least two "computationally independent" witnesses, then the WI property is nontrivial, and it is possible to infer WH from WI.

One example of problems with computationally independent witnesses is that of families of "claw-free" functions [16]. For these functions it is intractable (in nonuniform polynomial time) to find a claw: two arguments which map to the same image. One example of a claw free function is squaring modulo a composite. Finding a claw (two independent square roots of the same argument) implies factorization, which is assumed to be intractable. We call a claw free function *proper* if any image has at least two pre-images.

Theorem 4.1: Let G be a generator for a proper claw free function f, which generates pairs (x, w) where $x = f(w)$, with uniform distribution over the arguments w. Let (P, V) be a proof of knowledge system for proving knowledge of a pre-image of x. Then if (P, V) is WI over f, then it is WH over (f, G).

Proof(sketch): Assume that $V'_{P(x,w)}(x, y)$ can output a preimage of x with nonnegligible probability n^{-k}. We show how a polynomial time algorithm M can find "claws" in f with nonnegligible probability. M selects w' at random and computes $x' = f(w')$. Now it performs $V'_{P(x',w')}(x', y)$, using V' as a blackbox. This is possible since P is polynomial time. With probability n^{-k}, V' outputs a preimage of x'. Since the protocol is WI, and x' has at least two preimages, the probability this preimage differs from w' is at least $1/2$. Thus M finds a claw with nonnegligible probability, contradicting the claw-freeness of f. $\diamond$

Remark: Obviously, no single function is claw free with respect to nonuniform algorithms. The full proof of Theorem 4.1 involves the concept of a *family* of claw free functions, and is omitted.

Claw free functions are rare. Many candidates for intractable functions do not even have two preimages (e.g. the discrete log). We show here a transformation which transforms any relation R to a new relation R^2 for which each argument has two independent witnesses.

Given relation $R = \{(x, w)\}$, define R^2, where $((x_1, x_2), w) \in R^2$ iff $(x_1, w) \in R$ or $(x_2, w) \in R$.

Given a generator G for R, obtain a generator G^2 for R^2, by applying G twice independently, and discarding at random one of the two witnesses.

Theorem 4.2: Let G be a generator for relation R. Let (P, V) be a proof of knowledge system for R^2 (P proves knowledge of a witness of one of two instances in R). Then if (P, V) is WI over R^2, then it is WH over (R^2, G^2).

The proof of this Theorem is quite complicated, and is given in the Appendix. It is the only Theorem in this paper whose proof uses the concept of *witness extractor*, introduced in definition 4.2.

In typical cryptographic scenarios, it is assumed to be intractable to compute any witness from the common input alone. Under this assumption, the proof of Theorem 4.2 can be greatly simplified. Because of its cryptographic applications, we state this special case as a seperate Theorem.

Theorem 4.3: Let G be an invulnerable generator for relation R. Let (P, V) be a proof of knowledge system for R^2 (P proves knowledge of a witness of one of two instances in R). Then if (P, V) is WI over R^2, then it is WH over (R^2, G^2).

Proof: Assume the contrary. Then there exists V' whose probability of cracking an instance of R^2 after interacting with P is at least n^{-k}, for some integer k and infinitely many n. We construct M which has a non-negligible a-priori probability of cracking an instance of R (without interacting with P), thus contradicting G's invulnerability.

On input x, M uses G to generate an auxiliary solved instance (x_1, w_1). Now he uses the description of polynomial time P and his control over V' to run (P, V') on input (x, x_1), given in random order. M uses his knowledge of w_1 in order to perform P's part in the protocol. The probability that V' generates a witness to one of the two instances is n^{-k}. Because of the WI property, the witness V' produces is independent of the particular witness P is using, and so V' cracks x with probability $\frac{n^{-k}}{2} - \nu(n)$. This contradicts our assumption that G is an invulnerable generator. $\diamond$

In order to construct WI proofs of knowledge which are also WH, we composed two random instances of the NP language. This gives a new NP language, and thus has zero knowledge protocols (under cryptographic assumptions [13]). These protocols are also witness indistinguishable (Theorem 3.3). WI is preserved even if the basic steps are composed in parallel (Theorem 3.1). By Theorem 4.2, these parallel protocols are also witness hiding (on G^2).

Corollary 4.4 Let G be a generator for relation R. Then under the assumption that one way functions exist, R^2 has a constant round proof of knowledge which is witness hiding over (R^2, G^2).

Remark 1: If an NP problem has random self reducibility properties [2], then its respective relation R has perfectly zero knowledge proofs of knowledge which do not depend upon unproven cryptographic assumptions [20]. In this case, it is undesirable to go through the general reduction to an NP complete problem in order to construct a protocol for R^2. Fortunately, this is not necessary, and one can construct constant round perfectly witness indistinguishable proofs of knowledge for R^2 relations based on random self reducible languages (See [8] for an example based on the discrete log).

Remark 2: Note a subtle point in the argument preceding the corollary. The witness hiding property is not preserved under parallel composition (see for example Theorem 3.2), but witness indistinguishability is preserved. Consequently, in proving that the parallel composition is witness hiding, we first prove that the compostion is WI, and only then we deduce the WH property (with respect to generators of type G^2, which differ from the generators used in Theorem 3.2).

5 Applications

Identification Schemes: An identification scheme is a protocol which enables party P to prove his identity polynomially many times to party V without enabling party V to later misrepresent himself as P to someone else. Witness hiding proofs of knowledge are ideal candidates for identification protocols. The WH identification protocol never discloses P's witness, and nobody can misrepresent himself as P unless he really knows P's witness (since the protocol is a proof of knowledge). Furthermore, Corollary 4.4 shows a simple method of constructing constant round WH proofs of knowledge, and this can be used to limit the interaction between prover and verifier in identification protocols. One such identification scheme is described in [9]. It is based on the computational problem of squaring modulo a composite, which is assumed to be a claw free function. The original proof of security of the parallel version of this scheme involved detailed analysis of its number theoretic properties, but in fact the security of the protocol is just a special case of Theorem 4.1.

Constant round zero knowledge arguments: *Zero knowledge arguments* are zero knowledge proofs in which the soundness condition is required to hold only with respect to provers limited to random polynomial time computations ([4], [5]). The problem of constructing constant round zero knowl-

edge arguments (or preferably, proofs) for any language in NP was raised in [13], where a two round protocol for this problem was sketched. Proving the correctness of this protocol met unexpected technical difficulties, and this protocol was withdrawn. Goldreich and Kahan [11] modified this protocol, and obtained a three round (five messages) zero knowledge proof for any NP statement under the assumption that clawfree functions exist. A three round (six messages) perfectly zero knowledge argument for any NP statement was constructed in [5] under the assumption that *one-way group homomorphisms* exist (the intractability of the discrete log is a special case of this assumption). Using the concepts of witness hiding and witness indistinguishability, we construct a variety of constant round arguments for any NP language, including the only known construction of such protocols under the weak assumption that oneway functions exist.

Theorem 5.1:

1. Under the assumption that one-way functions exist, there exist three round (five messages) zero knowledge arguments for any NP statement.
2. Under the assumption that one-to-one one-way functions exist, there exist two rounds zero knowledge arguments for any NP statement. (This is optimal for any argument proven zero knowledge by blackbox simulation [12].)
3. Under the intractability of the discrete logarithm assumption, there exist two rounds perfectly zero knowledge arguments for any NP statement.

Our construction involves the concept of *trapdoor* commitment. A *trapdoor bit commitment scheme* is a regular commitment scheme with the additional property that B can construct commitments (indistinguishable from A's commitments) which B can later reveal in two possible ways: both as 0 and as 1. It was observed by several researchers, that using trapdoor commitment schemes, zero knowledge protocols can be performed in a bounded number of rounds. The problem in implementing this idea is that V has to prove to P in a constant number of rounds that he knows the trapdoor, without actually revealing it. Trying to make such a subprotocol ZK leads to circularity. This problem is solved by using WH protocols, since we do know how to construct constant rounds WH protocols (Corollary 4.4). Thus our bounded round protocol has the following two-phase structure:

1. V sends P a commitment scheme, and then proves by using a constant round WH protocol that he knows a trapdoor in it. P cannot learn the trapdoor because the protocol is WH. (P may learn other information from V, but this does not violate the ZK property of the full protocol, since the flow of information in this phase is from the verifier to the prover).
2. Using the trapdoor commitment scheme, P proves any NP statement in a bounded number of rounds.

The full protocol, including the construction of trapdoor commitment schemes from any one-way function, appears in [8].

Noninteractive proofs: Noninteractive zero knowledge proofs, as introduced in [3] and [7], postulate the existence of a publicly known random string (such as tables of random numbers prepared by the RAND corporation). Using this random string, the provers' goal is to write down proofs for NP statements, and these proofs should be verifiable by any verifier. A noninteractive proof is zero knowledge if the whole process of choosing a common random string and supplying a proof can be simulated in a polynomially indistinguishable way. General noninteractive zero knowledge (NIZK) protocols for any NP statement are constructed in [3] and [7] (under specific number theoretic assumptions) and recently in [18] (under the assumption that oneway permutations exist). However, a drawback of all the above schemes is that if the same common random string is used by more than $O(\log n)$ provers, then the zero knowledge property breaks down. We show that noninteractive witness indistinguishable (NIWI) protocols do not suffer from the same drawback. As in other parts of the paper, we consider only random polynomial time provers (with auxiliary input). (We note that in this case the [18] construction of NIZK requires a trapdoor to the oneway permutation.)

Definition 5.2: Noninteractive proof system (P, V) is *witness indistinguishable* over R if for any large enough input x, any $w_1, w_2 \in w(x)$, and for a randomly chosen public string σ, the ensembles $P(x, w_1, \sigma)$ and $P(x, w_2, \sigma)$, generated as P's proof are indistinguishable. The probability space is that of the random choices of σ together with P's random coin tosses. $\diamond$

Theorem 5.2: Any NIZK proof system is also a NIWI proof system.

We note a subtle point: Theorem 5.2 is not a special case of Theorem 3.3 ! Consider for example a distinguisher D, which for half of the choices of the public string σ is biased towards noninteractive proofs which use the witness w_1, and for the other half of the choices of σ is biased towards noninteractive proofs

which use the witness w_2. When averaging over all possible choices of σ, D has the same probability p of outputting 1, whichever of the two witnesses is used. Furthermore, it is possible that there exists a simulator M which produces strings on which D has probability p of outputting 1. Thus D may serve as a distinguisher which violates the WI property, without D violating the ZK property.

Proof (Theorem 5.2): Assume that for infinitely many triplets (x, w_1, w_2) of inputs together with their respective witnesses, D can distinguish between P using witness w_1 and P using witness w_2. Formally, for some $k > 0$:

$\sum_\sigma (|Prob(D(P(x, w_1, \sigma)) = 1)$
$-Prob(D(P(x, w_2, \sigma)) = 1)|) > n^{-k}$

where the sum is a weighted sum over the possible choices of σ, and the probabilities are taken over the random choices of P and D. We construct a nonuniform random polynomial time distinguisher D', which uses knowledge of both w_1 and w_2 to prevent the averaging effect described in the preceding discussion. On input z, a noninteractive proof for x using the public random string σ, D' generates n^{k+1} independent strings from each of the distributions $P(x, w_1, \sigma)$ and $P(x, w_2, \sigma)$ (once again, we note that this is possible because P is polynomial time, and D' can simulate P with the relevant auxiliary input). D' feeds these strings to D, and determines by a majority vote whether D is baised towards w_2 on σ. Then D' feeds D with z, and flips the decision made by D if and only if the bias test showed a bias towards w_2.

It is a simple matter to show that D' is biased towards w_1 on a random σ by at least $1/2n^k$. Now we can apply the proof of Theorem 3.3 to show that the original protocol was not zero knowledge. $\diamond$

Remark: The above proof uses the fact that P is polynomial time. It does not hold for exponential time provers. In fact, the truthful exponential prover in [18]'s protocol is deterministic (the only randomization is in the choice of σ), and so their protocol cannot be WI.

Theorem 5.3: Let (P, V) be a noninteractive proof system which is witness indistinguishable over R. Then the system remains witness indistinguishable even if polynomially many proofs are given using the same public random string σ.

The proof of Theorem 5.3 is similar to that of Theorem 3.1, and is ommitted.

Using Definition 4.2 as a definition of the witness hiding property for noninteractive proofs, we can prove a Theorem similar to Theorem 4.2:

Theorem 5.4: Let G be a generator for relation R. Let (P, V) be a noninteractive proof system for R^2 (P proves knowledge of a witness of one of two instances in R). Then if (P, V) is WI over R^2, then it is WH over (R^2, G^2).

Signature schemes: Bellare and Goldwasser [6] construct a signature scheme based on noninteractive ZK protocols. We modify this scheme by basing it on the concept of witness indistinguishability. Using our scheme, each user can securely sign polynomially many messages in a completely "memoryless" fashion, and any user can verify any signature. In contrast, current implementations of the BG scheme [6] fall into one of the following three categories: Either they are not memoryless (the signature of a message depends upon the number of previously signed messages), or signatures are not publicly verifiable, or coalitions of cheating users can forge signatures.

In our scheme, the trusted center publishes one random string R. We assume the existence of a secure commitment scheme E and a collection of pseudorandom functions $\{f_i\}$ [10]. Both assumptions follow from the assumption that one way functions exist ([19], [17]). The private key of each participant is (i, j), a pair of random indices of functions in $\{f_i\}$. The associated public key is $(E(i), E(j))$, a secure commitment to the private keys. A signature S of message m is a string $S(m) = (m, f_i(m), f_j(m), z)$, where z is a noninteractive witness indistinguishable proof that either $f_i(m)$ or $f_j(m)$ were computed correctly. The WI proof z uses the public random string R, and the publicly known $(E(i), E(j))$.

Theorem 5.5: The above signature scheme is not existentially forgeable under adaptive chosen message attack, even if polynomially many signatures use the same common random string R.

Proof(main idea): Assume that after requesting k signatures from P, V' can forge a signature for a new message M. From the soundness property of NIWI proofs, it follows that either the $f_i(M)$ part or the $f_j(M)$ part of the forged signature is computed correctly. W.l.o.g. assume $f_i(M)$ is. Using this, we contradict one of the two assumptions, that E is a secure commitment scheme, or that f_i is pseudorandom.

Assume a blackbox B_i which on input m outputs $f_i(m)$. Our goal is to obtain $f_i(M)$ without asking B_i for this specific value. Select j at random and compute $E(j)$. Now for each message m that V' sends, request the value of $f_i(m)$ from B_i, and construct the signature $(m, f_i(m), f_j(m), z)$ by computing the NIWI part of the signature using only the knowledge of j. Because of WI, V''s view after polynomially many signatures would be the same as if he was interacting with a real P (which may have used i in order to compute the NIWI part of the signature). Thus V' can still construct $S(M)$, and we can extract $f_i(M)$.$\diamond$

The complete proof of this Theorem will be given in the full version of this paper.

6 Conclusions

This paper analyses how the prover's knowledge might leak in interactive proofs. We study intermediate cases between the two extreme categories, of zero knowledge proofs, and of proofs which can leak everything. We show that there is an interesting and useful intermediate category: Proofs which are not zero knowledge, but which do not leak the prover's witness. We also initiate a case analysis of parallel composition of zero knowledge protocols: For statements which have only one witness, parallel composition may result in the disclosure of this witness, whereas for statements with two independent witnesses, this can not happen.

This paper also offers concepts which are possible alternatives to zero knowledge in the design of cryptographic schemes, especially when compositionality is required. As a methodology, it is convenient to think in terms of *Witness hiding* when doing the "high level" design of a cryptographic scheme, and to turn to the *witness indistinguishability* concept for the "low level" detailed proof of correctness of the scheme. However, these concepts are inadequete alternatives to zero knowledge when it is necessary to guarantee that the verifier does not learn how to perform any new computational task, and when the input statement has only one witness.

We conclude by listing a few topics for further research:

1. We described a particular methodology of constructing distributions of NP complete problems which have two independent witnesses (see Theorem 4.2). Do these problems have independent witnesses also under simpler distributions? For example, consider the probability space $G_{n,p}$ of graphs with n nodes in which each possible edge exists with probability p independent of the other edges. Are witness indistinguishable proofs of Hamiltonicity witness hiding over $G_{n,p}$?

2. Witness indistinguishable protocols are also witness hiding, under suitable conditions. Do they also hide partial information about the witnesses, such as individual bits?

3. All our constructions of WI protocols are modifications (e.g., parallelizations) of known zero knowledge protocols. Construct a WI (or WH) protocol, where the construction is based on a different idea.

4. Find other cryptographic applications for the new concepts.

Acknowledgements

We thank Mihir Bellare and Shafi Goldwasser for discussions concerning their signature scheme. Special thanks to Oded Goldreich for his colorful and very useful comments on an earlier version of this manuscript.

References

[1] M. Abadi, E. Allender, A. Broder, J. Feigenbaum, L. Hemachandra, *On Generating Solved Instances of Computational Problems* Proc. of CRYPTO88.

[2] D. Angluin, D. Lichtenstein, *Provable Security of Cryptosystems: a Survey* TR-288, Yale University, 1983.

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

[4] G. Brassard, D. Chaum, C. Crepeau, *Minimum Disclosure Proofs of Knowledge* JCSS, Vol. 37, 1988, pp. 156-189.

[5] G. Brassard, C. Crepeau, M. Yung, *Everything in NP can be argued in perfect zero-knowledge in a bounded number of rounds* Proc. of 16th ICALP, Stresa, Italy, 1989.

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

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

[8] U. Feige, A. Shamir, *Zero Knowledge Proofs of Knowledge in Two Rounds* Proc. of Crypto89.

[9] U. Feige, A. Fiat, A. Shamir, *Zero Knowledge Proofs of Identity* Journal of Cryptology, Vol 1, 1988, pp. 77-94. (Preliminary version in Proc. of 19th STOC 1987, pp. 210-217.)

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

[11] O. Goldreich, A. Kahan, *Using Claw-free Permutations to Construct Constant-Round Zero Knowledge Proofs for NP* in preparation.

[12] O. Goldreich, H. Krawczyk, *On Sequential and Parallel Composition of Zero-Knowledge Protocols* preprint, 1989.

[13] 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.

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

[15] 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.

[16] S. Goldwasser, S. Micali, R. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks* SIAM Journal on Computing, vol. 17, No. 2, pp. 281-308.

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

[18] D. Lapidot, A. Shamir, in preparation.

[19] M. Naor, *Bit Commitment Using Pseudorandomness* Proc. of CRYPTO89.

[20] M. Tompa, H. Woll, *Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information* Proc. 28th FOCS, 1987, pp. 472-482.

APPENDIX - Proof of Theorem 4.2

Let G be a generator for relation R, and let C denote algorithms which crack instances of R. Let C^2 denote algorithms which crack some instances of R^2 generated by G^2.

Main Lemma: There exists a uniform expected polynomial time algorithm C, such that for any C^2,

$Prob((x, C(x; C^2, G)) \in R) > p$

where $p = 1 - \sqrt{1-p'} - \nu(n)$, and p' is the probability that C^2 cracks a random instance of R^2. G and C^2 are given to C as blackboxes. In particular, C does not know which value of p it has to achieve.

Proof: We describe the behavior of C on input x of length n, generated by G:

Denote the following procedure by A: Apply $G(1^n)$ to obtain new instances x_i. Apply the pair-cracking algorithm C^2 to the pair (x, x_i) (given in random order). A succeeds if $R(x, C^2(x, G(1^n)))$ holds. Denote by A_k k successive independent applications of A. Intuitively, we would like C to be A_k, for some k polynomial in n. As we shall soon see, the construction of C cannot be that simple.

Let p_k denote the probability that A_k cracks x (for random x), where $p_0 = 0$. It is not trivial to compute this probability, since after A failed $k-1$ times, x is no longer from the original probability distribution G, but from G conditioned on $k-1$ previous failures.

Lemma 1: $p - p_{(k+1)} \leq (p - p_k)\frac{p+p_k}{2}$

Proof: By induction on k.

Base case: $k = 1$. A_1's cracking probability satisfies $p_1 \geq \frac{p'}{2} = \frac{2p-p^2}{2}$, which is also the probability obtained by substituting $p_0 = 0$.

Inductive step: For the sake of analysis we define $B_k(x, x_1)$ as an algorithm which executes $C^2(x, x_1)$, $A_k(x)$ and $A_k(x_1)$, and outputs whichever witnesses the three executions produce. $A_{(k+1)}$ can be viewed as an algorithm which picks just one random x_1, calls $B_k(x, x_1)$ just once, and checks whether B_k produced a witness for x. This is because the call that $B_k(x, x_1)$ makes to $A_k(x_1)$ becomes irrelevant, and so altogether B_k makes $k+1$ relevant calls to C^2.

The probability that both $A_k(x)$ and $A_k(x_1)$ produce a witness is p_k^2. Thus the expected number of entries (from x and x_1) to which B_k produces a witness, $E(B_k)$, satisfies:

$E(B_k) \geq 2p_k^2 + (2p - p^2 - p_k^2) = 2p - p^2 + p_k^2$

$= 2p - (p - p_k)(p + p_k)$

The desired result is obtained by noting that $p_{(k+1)} = \frac{E(B_k)}{2}$. $\diamond$

If $p < 1$ is a constant independent of n, then A_n succeeds with probability $p - \nu(n)$, because each application cuts down the distance to p by a constant fraction (at least by p). But if the unknown p is $1 - o(1)$, we do not know how many times to repeat the procedure. To keep the *expected* running time polynomial, we want to repeat the procedure $\frac{n}{1-p}$ times. This timer is chosen so that we do not spend too much time on instances which are not solvable, and so that we allow sufficient time to solve instances which are solvable.

Analysis of the success probability. By Lemma 1:

$$p - p_{\frac{k+1}{1-p}} \leq (p - p_{\frac{k}{1-p}}) \prod_{i=\frac{k}{1-p}}^{\frac{k+1}{1-p}-1} (\frac{p+p_i}{2})$$

$$\leq (p - p_{\frac{k}{1-p}})p^{\frac{1}{1-p}} \leq \frac{p - p_{\frac{k}{1-p}}}{e}$$

Thus $p_{\frac{n}{1-p}} \geq p - e^{-n}$.

The expected running time of this procedure can be shown to be polynomial. But how can C compute $\frac{n}{1-p}$? It may try to approximate $1-p$ probabilistically using $1 - p \simeq \sqrt{1-p'}$, where p' is the observed success rate of C^2. But this method has a serious flaw: We cannot hope to get a meaningful estimate of p' before we observe at least one failure. The expected time until a failure is encountered is $\frac{1}{1-p'}$, and for very small $1-p'$ this value can be much larger than $\frac{n}{1-p}$. Thus this method does not work in "real time".

In view of the above difficulty, we employ a different stopping procedure. It is based on an "online" version of the well known binary "knockout" tournament. The *players* are the auxiliary inputs x_i, generated by $G(1^n)$ in successive applications of A. A *game* between x_i and x_j is played by calling $C^2(x_i, x_j)$. The *winner* of the game is the input for which C^2 **failed** to find a witness. If C^2 finds a witness for both inputs, x_i is arbitrarily declared the winner. If C^2 fails completely and does not find any witnesses, this causes an *interrupt* in the tournament, and the tournament is stopped. Any player who ever loses a game is discarded from the tournament. The

scheduling of games among winners is performed by the following rule: Whenever there are two players who played the same number of games (having won all of them) they are paired for the next game. We do not care who wins the tournament - we are only interested in the timing of interrupts. Note that:

1. By the time m players are generated, at most $m-1$ games are played, since with each game one player is discarded. Each individual player plays at most $\log_2 m$ games.
2. Before the mth application of G (generation of the mth player) there are at most $\log_2 m$ active players in the tournament. (For any $0 \leq j \leq \log_2 m$, there is at most one player with j games).

Now we are ready to describe the complete expected polynomial time algorithm C. It involves three procedures, carried out in parallel, one step from each procedure at a time. The algorithm stops as soon as one of the procedures reaches its end.

Procedure 1: Use C^2 to crack x. Apply procedure A repeatedly, until $R(x, C^2(x, G(1^n)))$ holds (a witness for x is found).

Procedure 2: Exuastive search. Basic step - pick the next binary string y of length n and test $R(x,y)$. Procedure 2 ends when a witness for x is found. (At most 2^n steps).

Procedure 3: The online tournament. Basic step - receive a new player from Procedure 1. Complete all possible games to maintain the invariant that there are no two players with the same number of games. Each time an interrupt occurs, discard *all* players and start a fresh tournament (with 0 initial players). Procedure 3 ends when n interrupts are encountered.

Lemma 2: C finds a witness for x with probability $p - \nu(n)$.

Proof: It is sufficient to prove that with overwhelming probability, Procedure 3 does not stop in less than $\Omega(\frac{n}{1-p})$ steps. The result then follows from Lemma 1.

Procedure 3 is composed of n tournaments. Assume that there are $\frac{1}{2(1-p)}$ players in a tournament. They can play $\frac{1}{4(1-p)^2} = \frac{1}{4(1-p')}$ games among themselves (all possible pairs). The expected number of times C^2 fails to solve both instances in a pair is $(1-p')\frac{1}{4(1-p')} = \frac{1}{4}$. Thus the probability that among $\frac{1}{2(1-p)}$ players there exists a pair that C^2 does not solve is smaller than $1/4$. Since all tournaments are independent, the proof of the Lemma follows from the Chernoff bound.$\diamond$

Lemma 3: The expected running time of C is polynomial.

Proof(sketch): We prove a bound of $O(n^5)$ on the expected running time of C (tighter bounds can be obtained with more complicated arguments).

Let $S(T)$ denote the set of all inputs whose expected solution time using only Procedure 1 is at least T. Let q_T be the probability that a random input generated by G belongs to the set $S(T)$. If $\forall T \leq 2^n$, $Tq_T \leq n^4$, then the expected running time of C is at most n^5. Thus assume t is the minimal value for which $tq_t > n^4$. We prove that the contribution of the set $S(t)$ to the total expected running time of C is at most n^3.

After expected time $\frac{n^2}{q_t}$, Procedure 1 generates $\Theta(n^2)$ auxiliary inputs of set $S(t)$. Denote the set of these elements by S'. A *discard event* d_j for $x_j \in S'$ is the event that G produces x_k such that $C^2(x_j, x_k)$ produces a witness for x_j (and so x_j is discarded from the online tournament). The probability that the generation of x_k by G will result in an event d_j is at most $1/t$. Denote by d the event that x_k will cause at least one discard event in S'. Then $Prob(d) < \frac{n^2}{t}$. The probability that there are n events of type d in $\frac{n^2}{q_t}$ steps is thus at most $(\frac{n^2}{t})^n \binom{n^2/q_t}{n} < \frac{1}{n!}$.

Each discarder x_k can discard at most n players from the online competition, since this is the maximal number of games it plays. Thus if we have $o(n)$ discarders in the competition for the $\Omega(n^2)$ players of set S', $\Omega(n^2)$ of these players will remain in the competition, contradicting the fact that at most n players can remain in the online tournament. This implies that an interrupt event must have occurred. Thus the contribution of set t in a single competition is at most $n^2 + \frac{2^n}{n!}$, and n^3 for n competitions.$\diamond$

This completes the proof of the main Lemma. $\diamond$

Proof (Theorem 4.2): We have to prove that M has the same cracking probability as V'_P on (R^2, G^2). We prove the Theorem assuming (P,V) is perfectly witness indistinguishable. The case of computational indistinguishability requires more careful analysis.

Assume $V'_{P((x_1,x_2),w)}((x_1,x_2),y)$ has cracking probability p' on random instances of R^2. The construction of the main Lemma gives a cracking algorithm C which has cracking probability $p = 1 - \sqrt{1-p'} - \nu(n)$ of cracking random instances x of R. This C uses G to generate auxiliary inputs x_i, and uses the system (P,V') to crack the instances (x, x_i) of R^2. C can simulate the action of this system since he always knows w_i, a witness for x_i, and the protocol is WI. In order to achieve cracking probability p' on inputs $(x_1, x_2) \in R^2$, M calls $C(x_1)$ and $C(x_2)$, and the probability one of the two inputs is cracked is at least $2p - p^2 = p' - \nu(n)$.$\diamond$