One Message Proof Systems with Known Space Verifiers

Yonatan Aumann and Uriel Feige

Dept. Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel.

Abstract. We construct a proof system for any NP statement, in which the proof is a single message sent from the prover to the verifier. No other interaction is required, neither before nor after this single message is sent. In the "envelope" model, the prover sends a sequence of envelopes to the verifier, where each envelope contains one bit of the prover's proof. It suffices for the verifier to open a constant number of envelopes in order to verify the correctness of the proof (in a probabilistic sense). Even if the verifier opens polynomially many envelopes, the proof remains perfectly zero knowledge.
We transform this proof system to the "known-space verifier" model of De-Santis *et al.* [7]. In this model it suffices for the verifier to have space S_{min} in order to verify proof, and the proof should remain statistically zero knowledge with respect to verifiers that use space at most S_{max}. We resolve an open question of [7], showing that arbitrary ratios S_{max}/S_{min} are achievable. However, we question the extent to which these proof systems (that of [7] and ours) are really zero knowledge. We do show that our proof system is witness indistinguishable, and hence has applications in cryptographic scenarios such as identification schemes.

1 Introduction

We construct a proof system for the NP-language 3-SAT. The common input is a 3-CNF formula ψ. Prover P tries to convince the verifier V that ψ is satisfiable, without revealing additional information. The prover in our proof system need not be stronger than polynomial time, provided that he is given a satisfying assignment for ψ. We place a limitation on the space S of the verifier. Namely, $S_{min} < S < S_{max}$, where S_{min} is the amount of space that suffices in order to verify the proof, and S_{max} is a bound on the space used by V, so that if $S < S_{max}$ then V does not learn "too much" (in a sense to be defined shortly). We stress that S is polynomial in the input length. The proof itself is given as one message sent from P to V. There is no interaction between P and V other than this one message sent by P.

Our work was motivated by the work of De-Santis, Persiano and Yung [7]. They construct such a proof system for an NP-complete language. An open question that they pose concerns the ratio S_{max}/S_{min}, known as the *tolerance* of the proof system. In their proof system, S_{max}/S_{min} is bounded by 2. Hence, the proof system can be employed only if the space of the verifier can be characterized within a very narrow range (not only the space of the truthful verifier, but also the space available to potential cheating verifiers). De-Santis *et al.* ask whether the tolerance of such proof systems can be improved. We answer this question in the affirmative. We construct a proof system scheme, parameterized by k, where k may be polynomially related to the length of ψ. For any desired value of k, our proof system scheme gives a proof system for 3-SAT with tolerance $S_{max}/S_{min} = k$.

Our original intention was to show that our proof system is zero knowledge (see Definition 2). However, we could not prove this. We could not prove the zero knowledge property even for the protocol in [7]. We will return to this issue in Sect. 1.4. Falling short of proving the zero knowledge property, we show that our proof system is *witness indistinguishable*. This property, which is weaker than zero knowledge, suffices for some cryptographic applications (see Sect. 1.5). It remains a major open question if a one message proof systems can be zero knowledge with respect to a known space verifier (under some reasonable definition of the concept of zero knowledge).

1.1 Definitions and Statement of our Results

The output of an algorithm A on input x is denoted by $A(x)$, which can be a random variable if A is a randomized algorithm.

Let $R = (x, w)$ be a relation testable in polynomial time in which the sizes of x and w are polynomially related, and let L_R be the NP-language associated with it. (E.g., x may be a satisfiable 3-CNF formula, w its satisfying assignment, and L_R the language 3-SAT.) In a proof system (P, V) both prover P and verifier V are probabilistic polynomial time. They both see a common input x, for which P tries to convince V that $x \in L_R$. The truthful prover is given a witness w such that $(x, w) \in R$ as auxiliary input. The size S of the work space available to the verifier is known to satisfy $S_{min} < S < S_{max}$. The verifier V may also have auxiliary input y of polynomial length (e.g., leftover information from executions of previous protocols). We do not require that $|y| < S_{max}$.

Definition 1. Language L_R has a *one message proof system* (P, V) with error ϵ if the following holds:

1. *Completeness:* the truthful prover can convince the truthful verifier to accept true statements. If $(x, w) \in R$, then $V(x, P(x, w))$ accepts.
2. *Soundness:* even a computationally unbounded cheating prover has only small probability of convincing the truthful verifier to accept false statements. If $x \notin L$, then for any message m, $Pr[V(x, m) = accept] \leq \epsilon$.

The proof system is a *proof of knowledge* if there exists a polynomial time "knowledge extractor" M such that for any x and m, if $Pr[V(x, m) = accept] > \epsilon$, then $(x, M(x, m)) \in R$.

In the following definitions of *zero knowledge* and *witness indistinguishability*, one may assume that at the end of the protocol the verifier outputs the contents of its work space (it "dumps" its memory).

Definition 2. A one message proof system for L_R is *perfectly (statistically, computationally) zero knowledge* if for any (possibly cheating) verifier V with work space $S < S_{max}$, there exists an expected polynomial time simulator M, such that for any $(x, w) \in R$, and any auxiliary input y, the distributions $V(x, y, P(x, w))$ and $M(x, y)$ are perfectly (statistically, computationally) indistinguishable.

Definition 3. A one message proof system for L_R is *perfectly (statistically, computationally) witness indistinguishable* if for any (possibly cheating) verifier V with work space $S < S_{max}$, for any $x \in L_R$, for any witnesses w_1 and w_2 that satisfy $(x, w_1) \in R$ and $(x, w_2) \in R$, and any auxiliary input y, the distributions $V(x, y, P(x, w_1))$ and $V(x, y, P(x, w_2))$ are perfectly (statistically, computationally) indistinguishable.

Theorem 4. *For any polynomial ratio $S_{max}/S_{min} = k$, there is a statistically witness indistinguishable one message proof of knowledge for 3-SAT (and hence for any NP-statement) with tolerance k.*

We remark that theorem 4 requires no cryptographic assumptions.

1.2 Related Work

Interactive proof systems and the concept of zero knowledge were introduced by Goldwasser *et al.* [14]. In the [14] model the verifier is probabilistic polynomial time, with no restrictions on its space. In Goldreich *et al.* [13] and Brassard *et al.* [4] it was shown that under certain computational complexity assumptions, all NP-languages have zero knowledge interactive proofs. Interaction seems to be an essential ingredient for zero knowledge. Goldreich and Krawczyk [15] prove that at least two rounds of messages are required for zero knowledge proofs for languages not in BPP, if the zero knowledge property is proved by *blackbox simulation*. The need for interaction can be replaced by the assumption that the prover and verifier share a common random string. In this model, non-interactive zero knowledge proof systems can be constructed under certain cryptographic assumptions [3, 12].

The study of interactive proof systems with space bounded verifiers was initiated by Condon [5]. Dwork and Stockmeyer [8] studied zero knowledge aspects when the verifier is a finite automaton. Kilian [17] constructed a proof system for any language in PSPACE, which is zero knowledge with respect to a log-space verifier. The model studied in our paper, that of one message known space verifier, was introduced by De Santis *et al.* [7]. Its goal was to achieve zero knowledge in one message, no cryptographic assumptions, and with respect to reasonably strong verifiers. It turns out that the main issue concerned is obtaining a one message proof, since Kilian's protocol uses interaction extensively, but does not require cryptographic assumptions, and can easily be adapted to the scenario of known space verifier with polynomial space bounds.

1.3 Main Ideas in our Construction

It is convenient to consider the following envelope scenario. An *envelope* is an idealized version of *bit commitment*. The prover can commit to a bit by placing it inside an envelope, and sealing the envelope. Thereafter, the prover cannot change the value of the bit. The verifier cannot see the value of the committed bit until she explicitly opens the envelope.

In the envelope scenario, in order to convince the verifier that $x \in L_R$, the prover sends his proof hidden in a sequence of ℓ sealed envelopes. If the prover is truthful, then whichever envelopes the verifier chooses to open, the verifier accepts. If $x \notin L_R$, then it suffices for the verifier to open ℓ_{min} envelopes (where $\ell_{min} < \ell$) in order to have non-negligible probability of rejecting. The proof is zero knowledge with respect to verifiers that open no more than ℓ_{max} envelopes (where $\ell_{min} < \ell_{max} < \ell$). The tolerance of the proof system is ℓ_{max}/ℓ_{min}.

We first construct a zero knowledge proof system for 3-SAT in the envelope scenario. Our construction is based on the concept of *randomized tableau*, first introduced by Kilian [16]. We modify his original constructs so as to get control of the tolerance of the proof system, and so as to improve efficiency. The details of our protocol appear in Sect. 2.

Once we have a protocol in the envelope scenario, we transform it to a protocol in the known space scenario. For this we replace the ideal bit commitments based on envelopes by a computational form of bit commitment based on *inner products*, as introduced by Kilian [17] and modified by De Santis *et al.* [7]. The idea is as follows. Let S_{min} denote the minimum space of the verifier, and let $b \simeq S_{min}/\ell_{min}$. P commits to the value of a bit z_i by selecting two vectors $v_1(z_i), v_2(z_i) \in \{0,1\}^b$ at random, subject to $v_1(z_i) \bullet v_2(z_i) = z_i$, where $\bullet$ denotes the inner product operation (the inner product of two vectors is the *sum mod 2* of the *and* of the respective bits). In order to send the sequence of ℓ bits $\{z_1, z_2, ..., z_\ell\}$, the prover first sends the sequence $v_1(z_1)$, $v_1(z_2)$, ... , $v_1(z_\ell)$, and then the sequence $v_2(z_1)$, $v_2(z_2)$, ... , $v_2(z_\ell)$. The verifier can "open" ℓ_{min} committed bits by first saving their respective v_1 vectors (this requires space roughly $b\ell_{min} \simeq S_{min}$), and then performing the inner products with the respective v_2 vectors online. Intuitively, the desired property that the verifier cannot open more that ℓ_{max} of the committed bits follows from the fact that in space $S_{max} \simeq b\ell_{max}$ the verifier cannot save more than ℓ_{max} of the v_1 vectors, and hence lacks sufficient information to recover the values of $\ell_{max} + 1$ bits at the time that the v_2 vectors arrive.

1.4 The Problems with Zero Knowledge

There is no problem in showing that the protocol in the envelope model is zero knowledge. This is true also of the protocol constructed in the [7] paper. Hence intuitively, it seems that also the protocol in the known space model is zero knowledge. This intuition is supported by the following communication complexity game. Player A receives in private a random vector $v_1 \in \{0,1\}^b$ and player B receives in private a random vector $v_2 \in \{0,1\}^b$. Player A can send a message of s bits to B. How large should s be so that the probability that player B computes $z = v_1 \bullet v_2$ is significantly greater than 1/2? Clearly, if $s = b$, then A can send v_1 to B, and B can compute z. However, if s is significantly smaller than b, then B's probability of guessing the value of z is essentially 1/2, as proved in [6]. Returning to our scenario of known space verifier, this implies that the verifier needs to store almost b bits of information regarding the vector $v_1(z_i)$, if she is later to open the committed bit z_i. Hence if the verifier's space is limited to S_{max}, it seems that she cannot store sufficient information in order to recover more than ℓ_{max} of the committed bits.

The above intuitive argument was formalized in [7] in the following way. They considered a game in which each player receives k random vectors (each of b bits), player A sends a single message of s bits, and player B has to compute the value of the k respective inner products. [7] prove that unless $s \simeq kb$, there is only negligible probability that B computes correctly all k inner products. From this [7] conclude that the verifier cannot recover more then ℓ_{max} of the committed bits, which would seem to imply that the proof system is zero knowledge in the known space model. However, this line of argument does not address the following issues:

1. It still has to be established that in order for the verifier to get meaningful information, she must explicitly open committed bits. Perhaps after seeing P's proof, the verifier can output the lexicographically first satisfying assignment for ψ, without outputting any of the committed bits z_i. (Its hard to imagine how such a thing can be done, but it was not shown that it cannot be done.)
2. There is side information available to the verifier. The committed bits z_1, ... ,z_ℓ, encode a satisfying assignment for ψ. They are not truly random and independent bits and this causes dependencies among the vectors $v_1(z_1)$, ... ,$v_1(z_\ell)$, $v_2(z_1)$, ... , $v_2(z_\ell)$.

We were unable to complete the argument to a full rigid proof that the protocol is zero-knowledge. The following are the types of problems we have encountered.

1. Communication complexity arguments alone do not seem to suffice, as they only address the space of the verifier. The verifier has polynomial space, and this suffices for her in order to *find* a satisfying assignment for ψ. Thus obviously, communication complexity arguments cannot exclude the possibility that V eventually outputs a satisfying assignment for ψ.
2. If one attempts to construct a simulator for V, then one encounters the following problem. In all previous protocols that we know of (e.g [14]), it is clear from the protocol which of the committed bits are revealed. This fact is later used in constructing the simulation. But in our model, there is no indication of which bit commitments the verifier chooses to open (if any).

3. The most successful way of proving that a protocol is zero knowledge is by *blackbox simulation*, in which the simulator M treats the possibly cheating verifier as a blackbox, and studies its input/output relations at intermediate steps of the protocol. However, a one message proof system leaves no room for blackbox simulation. In fact, it can be shown that only languages in BPP have a one message proof system that is blackbox-simulation zero knowledge. We know of only one case in which something different from blackbox simulation was used in order to prove that a protocol is zero knowledge. This was done in an interactive scenario where both verifier and simulator where restricted to logarithmic space [17], but does not seem to apply in our context.

We regard it as a highly challenging open question to prove that a one message proof system is zero knowledge, with respect to some reasonable definition of zero knowledge.

1.5 Witness Indistinguishability

We prove that our protocol is witness indistinguishable (in the known space model). Our proof is based on communication complexity arguments, and takes into account the dependencies between the committed bits. See Sect. 4 for details.

The concept of witness indistinguishability first received comprehensive treatment in [11] (though it was used implicitly also in earlier works). It turns out that if certain easy to meet restrictions are placed on the distribution of inputs to the protocol, then any witness indistinguishable protocol is also *witness hiding* - at the end of the protocol the verifier cannot compute any witness to the input statement, unless she could do so before the protocol began. Witness hiding is a natural property to consider in the context of proofs of knowledge. If a proof of knowledge is witness hiding, then the verifier of the proof cannot use it to become a prover later, since becoming a prover requires knowledge of a witness to the input statement. Witness hiding proofs of knowledge can serve as identification schemes in the spirit of [9]. For more information on the theory of witness indistinguishability and witness hiding see [11], and for applications see [10, 12].

2 The Protocol

Consider a predicate $\psi \in$ 3–CNF. Prover P claims he has a satisfying assignment w for ψ, and wishes to prove this to the verifier V. We describe the protocol for doing so in several stages (the protocol we describe is a simplified version of the protocol, a somewhat more efficient version of the protocol is noted upon in Sect. 3 and will appear in the full version of our paper). First we describe a transformation of ψ into a different predicate Ψ, where each variable is represented by K variables, where $K = \Theta(k)$ (k being the tolerance). Then we convert this predicate into a *Permutation Branching Program (PBP)* format. Next we construct a *Random Computation Tree (RCT)* for the PBP, to compute the value of the PBP, while hiding intermediate results. Next we describe the protocol in the pure envelope scenario, and show how the prover can prove his claim, with no interaction, and zero-knowledge. The final stage, the implementation of envelopes in the non-interactive bounded-space scenario, was described in Sect. 1.3.

2.1 Splitting the Variables

Let

$$\psi = \bigwedge_{i=1}^{m} C_i = \bigwedge_{i=1}^{m} (L_{i,1} \vee L_{i,2} \vee L_{i,3}) ,$$

where

$$L_{i,l} \in \{x_j\}_{j=1}^{T} \cup \{\bar{x_j}\}_{j=1}^{T} .$$

We replace each variable x_j of ψ by the *exclusive "or"* of K new variables. For each x_j of ψ, set $x_j = \bigoplus_{q=1}^{K} y_j^q$, *symbolically*, and set $\bar{x_j} = \bar{y}_j^1 \oplus (\bigoplus_{q=2}^{K} y_j^q)$. Set

$$\Psi = (\bigwedge_{i=1}^{m_1} C_i')$$

where C_i' is the symbolic representation of C_i with each literal replaced by its symbolic representation as an exclusive "or" of K variables.

The new predicate Ψ has a satisfying assignment iff ψ has one. Furthermore, any satisfying assignment for Ψ induces one for ψ, and any satisfying assignment for ψ induces several ones for Ψ, in a natural way.

2.2 Permutation Branching Program Format

Next, we describe how to transform each clause of Ψ into a *Permutation Branching Program (PBP)* format. For the general case, Barrington [2] describes such a transformation yielding polynomial size programs with permutations in S_5. For our case, however, we can achieve more efficient representation, with permutations in S_3 and linear size programs.

A 3–PBP, B, of length l, over the set of Boolean variables Y, is an ordered list of triplets

$$B = ((v^1, \sigma_0^1, \sigma_1^1), \ldots, (v^l, \sigma_0^l, \sigma_1^l)),$$

where $v_i \in Y$, and σ_0^i, σ_1^i, are permutations in S_3. For a given assignment $f : Y \to \{0, 1\}$, the program B *yields* the value

$$\mathrm{val}(B, f) = \prod_{i=1}^{l} \sigma_{f(v^i)}^i .$$

We say that B *accepts* f if $\mathrm{val}(B, f) = e$, where e is the identity permutation.

Given a clause C'_i we show how to transform it into a PBP $B(C'_i)$ such that $B(C'_i)$ accepts f iff f is a satisfying assignment. Furthermore if it does not accept, then it yields some fixed permutation π.

Set

$$\pi_1 = (1\ 2)(3), \pi_2 = (1)(2\ 3), \pi_3 = (1\ 3)(2)$$

For $C'_i = (L'_{i_1} \vee L'_{i_2} \vee L'_{i_3})$, and $p = 1, 2, 3$, define B_i^p of length K

$$B_i^p = ((y_{i_p}^1, \pi_p, e), (y_{i_p}^2, e, \pi_p), \ldots, (y_{i_p}^{K-1}, e, \pi_p), (y_{i_p}^K, e, \pi_p))$$

if L'_{i_p} corresponds to a positive occurrence of variable x_{i_p}, and

$$B_i^p = ((y_{i_p}^1, e, \pi_p), (y_{i_p}^2, e, \pi_p), \ldots, (y_{i_p}^{K-1}, e, \pi_p), (y_{i_p}^K, e, \pi_p))$$

if L'_{i_p} corresponds to a negative occurrence of variable x_{i_p}.

Define

$$B(C'_i) = B_i^1 \circ B_i^2 \circ B_i^1 \circ B_i^2 \circ B_i^3 \circ B_i^2 \circ B_i^1 \circ B_i^2 \circ B_i^1 \circ B_i^3$$

where $B \circ B'$ is the concatenation of B and B'.

We claim $B(C'_i)$ is the desired PBP.

Lemma 5. *Let f be an assignment to the $y_{i_p}^q$'s, then*

$$\mathrm{val}(B(C'_i), f) = \begin{cases} e & f \textit{ satisfies } C'_i \\ \pi & \textit{otherwise} \end{cases}$$

where $\pi = (1\ 3\ 2)$.

Proof: Consider B_i^1. By construction,

$$\mathrm{val}(B_i^1, f) = \begin{cases} \pi_1 & y_{i_1}^1 \oplus y_{i_1}^2 \oplus \cdots \oplus y_{i_1}^K = 0 \\ e & y_{i_1}^1 \oplus y_{i_1}^2 \oplus \cdots \oplus y_{i_1}^K = 1 \end{cases}$$

The same holds for B_i^2 and B_i^3, with π_2 and π_3. Calculation now shows that if any of the B_i^p's yields e then so does the full $B(C'_i)$, and otherwise $B(C'_i)$ yields π. ∎

Clearly, the representation of $B(C'_i)$ is linear in that of C'_i. In fact the representation is rather concise: for a clause C'_i of length $3K$, $B(C'_i)$ has $10K$ entries.

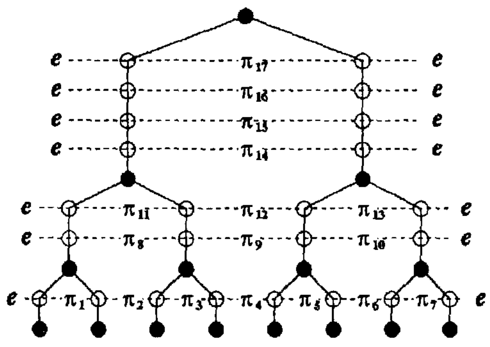

Fig. 1. An RCT for $t = 8$

2.3 Randomized Computation Trees

Consider a PBP, B. For a given assignment f, we have $\mathrm{val}(B, f) = \prod_{j=1}^{t} \sigma_j$ (where the σ_j depends on the assignment f, and the PBP B). W.l.o.g. assume t is a power of 2. Define the following tree structure: take a full binary tree with t leaves, and replace each node of height l by a chain of length $2^l + 1$. The root node is not replaced. A picture of this structure for $t = 8$ is depicted in Fig. 1. Call the lowest node of each chain a *combining node*, and the other nodes *chain nodes*. To the right and the left of each chain node we place a permutation. Neighboring nodes share permutations (see Fig. 1). We call these permutations *hiding permutations*. For a node, w, denote by $LP(w)$ and $RP(w)$ its left and the right hiding permutations, respectively. The outer permutations, on each side, are fixed to be the identity permutation (e). Other permutation are chosen randomly.

We define the values at the nodes of the computation tree recursively. For a combining node w, denote its left and right children by w_L and w_R, respectively. For a chain node w, denote its child by w_C. The value of a node is defined:

$$\mathrm{val}(w) = \begin{cases} \sigma_i & w \text{ is the } i\text{-}th \text{ leaf.} \\ \mathrm{val}(w_L) \cdot \mathrm{val}(w_R) & w \text{ is a combining node.} \\ \mathrm{val}(v) = LP(w)^{-1} \cdot \mathrm{val}(w_C) \cdot RP(w) & w \text{ is a chain node.} \end{cases} \quad (1)$$

For a given permutation branching program B, assignment f, and set of hiding permutations R, we denote the corresponding *Random Computation Tree (RCT)*, by $\mathcal{T} = \mathcal{T}(B, f, R)$.

For a node w, let R-path(w) be right–most path leading from the node down to a leaf, and let L-path(w) be the left–most path leading from a leaf to the node. Let Span(w) be the sequence of leaf nodes under w, in order from left to right. Then

$$\mathrm{val}(w) = \prod_{u \in \text{L-path}(w)} LP(u)^{-1} \cdot \prod_{u \in \mathrm{Span}(w)} \mathrm{val}(u) \cdot \prod_{u \in \text{R-path}(w)} RP(u),$$

where all products are taken in order. Thus, the Random Computation Tree is a computation tree for B, with intermediate values "padded" by random permutations. Specifically, at each level of chain nodes the value is padded by one additional permutation from the right, and one from the left. The root node is padded only by the identity permutations, and hence,

$$\text{val}(\text{root}) = \text{val}(B, f).$$

All other nodes are padded by random permutations.

2.4 The Protocol in The Pure Envelope Model

Let w be the witness available to P. Formally $w : \{x_j\} \to \{0,1\}$ is a truth assignment to the variables of ψ. For each variable x_j, P chooses at a random truth assignment, denoted by f, for $(y_j^1, \ldots, y_j^K)$, subject to $\bigoplus_{q=1}^{K} f(y_j^q) = w(x_j)$. Next, for each PBP $B(C_i')$, P chooses at random a set of hiding permutations, and prepares the corresponding RCT $\mathcal{T}_i$. In all appearances of a variable x_j, the *same* values for the y_j^q's are being used. Next, for all the y_j^q's, prover P puts the assignment values for these variables in separate envelopes. In addition, for each RCT, $\mathcal{T}_i$, corresponding to the PBP $B(C_i')$, P puts the value of each interior node in a separate envelope, as well as the set of all hiding random permutations, each permutation in a separate envelope.

All envelopes are now sent to the verifier. The verifier can open between four and $K-1$ envelopes. The verifier performs a *correctness test*:

Correctness test: The verifier chooses at random one RCT $\mathcal{T}$, and one non-leaf node $w \in \mathcal{T}$. For this node the verifier checks that:

1. The value val(w) was correctly constructed. V opens the related envelopes and checks that val(w) satisfies eq. (1).
2. If w is the root node then val(w) $= e$ (implying that the corresponding clause is satisfied).

If any of these tests fail, the verifier rejects, otherwise, she accepts.

The above envelopes contain permutation in S_3. Each permutation can be represented by 3 bits. Confining ourselves to single-bit envelopes, we have the prover send each permutation in a sequence of 3 envelopes. We assume the verifier can open between $\ell_{min} = 12$ and $\ell_{max} = K - 1$ of these envelopes[1]. The tolerance of the system is $k = K/12$.

Lemma 6. *The above protocol is a perfectly zero knowledge proof of knowledge for 3-SAT in the envelope model.*

[1] Kilian (private communication) remarks that ℓ_{min} can be reduced to 3. However, for a given tolerance, this reduction entails a degradation in the error probability and complexity. Consequently, this does not provide a more efficient protocol in the known space scenario.

The proof of this lemma is omitted due to space limitations. It is based on the construction of a simulator M that sends "empty" envelopes to V. For any set of less than K envelopes that V chooses to open, M supplies "contents" to these envelopes, with a distribution that is perfectly indistinguishable from the distribution that would arise if a real prover was sending the envelopes.

The envelope protocol is transformed into a protocol for the bounded space scenario as described in Sect. 1.3.

3 Cheating Probability

Suppose $\psi \notin$ 3-SAT. The proof-message sent by P, induces an assignment $w : \{x_j\} \to \{0,1\}$ of the variables of ψ. Let C_i be a clause not satisfied by w (there must be at least one such clause). Suppose that for the correctness test, verifier V chooses to test $\mathcal{T}_i$ (the RCT associated with C_i). Then there is at least one node $w \in \mathcal{T}_i$ for which the test fails. The number of nodes in an RCT is $\leq 20K \log K = O(k \log k)$, where k is the tolerance. Thus, for a given clause C_i, not satisfied by w, Pr[correctness test fails] $\geq 1/O(k \log k)$. Let m be the total number of clauses in ψ, and let $\hat{m}$ be the maximum number of clauses which can be satisfied simultaneously. We obtain

$$p = \Pr[V \text{ accepts}] \leq 1 - \left(\frac{m - \hat{m}}{m} \cdot \frac{1}{O(k \log k)}\right). \qquad (2)$$

For any $\psi \notin$ SAT this probability is $\leq (1 - 1/O(nk \log k))$. By [1], for $\psi \in$ MAX-SNP, the fraction $m/(m - \hat{m}) = O(1)$, and hence the probability in (2) gives $\leq (1 - 1/O(k \log k))$. The error probability can be further reduced by sequential repetition of the protocol. With t sequential repetitions the acceptance probability decreases to p^t. Clearly, the repeated protocol still remains a one message proof. Note that in the envelope model sequential repetition requires that the bounds on the number of envelopes V can open ($\ell_{min} \leq i \leq \ell_{max}$) must hold for each repetition separately. For the known space scenario, this automatically holds.

Finally we note on a method to further reduce the cheating probability (and consequently the complexity of the protocol for any fixed security parameter). Consider again the envelope model, and suppose that the RCT's are sent one by one, and that for *each* RCT, verifier V can open between 12 and $K - 1$ envelopes. In this case V can test each RCT separately. Thus, for *each* non-satisfied clause, C_i, V will detect that C_i is not satisfied with probability $\geq 1/O(k \log k)$. Thus,

$$\Pr[V \text{ accepts}] \leq \left(1 - \frac{1}{O(k \log k)}\right)^{(m - \hat{m})}.$$

However, this later protocol, as described above, is not zero-knowledge, even in the pure envelope setting (a sequence x_j may be sent many times, until all its y_j^q's are revealed). In order to preserve the zero-knowledge property, further modification to the construction are introduced. We omit the details here. When proving witness indistinguishability, this modified protocol introduces extra complexities in the analysis. The proof we give in the next section is for the original protocol.

4 Proof of Witness Indistinguishability

We now sketch the proof that the proof system is statistically witness indistinguishable in the known space model.

As corollary of Lemma 6 and of the theory of witness indistinguishability [11], we have:

Corollary 7. *The proof system is perfectly witness indistinguishable in the envelope model.*

We shall use the above corollary in our analysis of the known space model. We use the following notation:

n - length of input statement.
w_1, w_2 - possible witnesses to the input statement.
ℓ - number of envelopes.
ℓ_{min} - number of envelopes that V needs to open.
ℓ_{max} - number of envelopes that V is allowed to open.
b - number of bits per envelope in the inner product representation.
S - space of verifier.
S_{min} - space guaranteed to be available to V. We assume $S_{min} > n$.
S_{max} - space guaranteed not to be available to V.
$Z = z_1, z_2, ..., z_\ell$ - message sent by P in envelope model.
R - number of possible values of Z consistent with a single witness.
$v_1(z_i)$, $v_2(z_i)$ - vectors of b bits, $v_1(z_i) \bullet v_2(z_i) = z_i$.
m - message sent by P in the known space model.
m_1 - first part of message (only the v_1 vectors).
m_2 - second part of message (only the v_2 vectors).

In the envelope model, all messages Z consistent with a specific witness are equi-probable. We require that for every z_i, $v_1(z_i) \neq 0^b$.

The protocol is repeated sequentially many times, with independent random bits for the prover, so as to decrease the error probability. It suffices to prove that a single iteration is witness indistinguishable, since in our model we allow the verifier to have auxiliary input, and hence witness indistinguishability is preserved under repetition.

Consider any two witnesses w_1 and w_2, and construct the following matrix M. The rows of M are labeled by messages m_1 and the columns by messages m_2. Hence M has $(2^b - 1)^\ell$ rows and $2^{b\ell}$ columns. For each entry M_{ij} interpret i and j as m_1 and m_2 respectively, and set $M_{ij} = 1$ if i, j is a message that corresponds to P using w_1, $M_{ij} = -1$ if i, j is a message that corresponds to P using w_2, and $M_{ij} = 0$ otherwise. Then each row of M has exactly $R2^{-\ell}2^{b\ell}$ entries that are 1, and $R2^{-\ell}2^{b\ell}$ entries that are -1. If P is using w_1 (w_2, respectively) as a witness, then in effect his message m corresponds to a random 1 entry (-1 entry, respectively) in M.

We model the scenario as a communication complexity game. $m = (m_1, m_2)$ is picked with uniform probability from the nonzero entries of M. Alice sees m_1 and Bob sees m_2. Alice sends to Bob a message of length S. Bob outputs either

1 or -1. We want to bound $Pr[B(m) = M(m)]$. The probability is taken over the random choice of $m = (m_1, m_2)$. (The optimal strategy for Alice and Bob is deterministic.)

Lemma 8. *If in the communication game $Pr[B(m) = M(m)] = 1/2 + o(n^{\omega(1)})$, then the protocol is statistically witness indistinguishable.*

Proof: V's behavior can be simulated in the communication complexity game. Alice simulates $V(m_1)$, then sends the contents of the workspace of V to Bob (at most S bits), and Bob completes the simulation on m_2. If V can distinguish between w_1 and w_2, then so can Alice and Bob. ▮

It remains to analyze the communication complexity game. Alice sends one of 2^S possible messages. This partitions the rows of M into 2^S horizontal blocks. Bob holds a column. We need to prove that with high probability, this column is balanced along the block (sum of entries is approximately 0).

A block is *heavy* if it contains at least $2^{b\ell - S - \alpha}$ rows. The probability that a random row is in a heavy block is at least $1 - 2^{-\alpha}$ (ignoring the negligible factor of $(\frac{2^b-1}{2^b})^\ell$ due to the fact that $v_1 \neq 0^b$). We show that for heavy blocks, most columns are balanced.

Consider an arbitrary row r, and a random column c. Then

$$Pr[M(r,c) = 1] = Pr[M(r,c) = -1] = R2^{-\ell}$$

Consider now a heavy block B. Take a random column c and consider the random variable $X_c = \sum_{r \in B} M(r,c)$. Since $Pr[M(r,c) = 1] = Pr[M(r,c) = -1]$, then the expectation satisfies $E[X_c] = 0$. To show that most columns are nearly balanced we will bound $var[X_c]$, and use Chebyshev's inequality, $Pr[|X - E[X]| > \gamma] \leq \frac{var[X]}{\gamma^2}$.

$var[X_c] = E[(X - E[X])^2] = E[X^2] = \sum_{r_1, r_2} E[M(r_1,c)M(r_2,c)]$. To bound this last sum we use the fact that the protocol is perfectly witness indistinguishable in the envelope model, and hence knowledge of the contents of ℓ_{max} envelopes gives no information on whether P is using w_1 or w_2.

Consider two rows $r_1, r_2 \in B$. Each string r_i is composed of ℓ vectors, each of b bits. It follows from the witness indistinguishability property in the envelope model that:

Lemma 9. *If r_1 and r_2 agree on no more than ℓ_{max} matching vectors, then $E[M(r_1,c)M(r_2,c)] = 0$.*

Hence

$$\sum_{r_1, r_2 \in B} E[M(r_1,c)M(r_2,c)] \leq 2R2^{-\ell}|B| \sum_{j=\ell_{max}+1}^{\ell} (\binom{\ell}{j} 2^{-bj} 2^{\ell b})$$
$$< 4R2^{-\ell}|B| \binom{\ell}{\ell_{max}+1} 2^{-b(\ell_{max}+1)} 2^{\ell b}$$

Using $4R2^{-\ell}\binom{\ell}{\ell_{max}+1} < 1$ we obtain that $var[X_c] < |B|2^{-b\ell_{max}}2^{b\ell}2^{-b}$. By Chebyshev inequality, $Pr[X_c > \gamma|B|] \leq 2^{b\ell}/2^{b\ell_{max}}2^{b}\gamma^2|B|$.

Call a column *biased* if $X_c > \gamma|B|$. There are at most $2^{2b\ell}/2^{b\ell_{max}}2^{-b}\gamma^2|B|$ biased columns. The total number of nonzero entries in biased columns is at most $2^{2b\ell}/2^{b\ell_{max}}2^{b}\gamma^2$. The total number of nonzero entries in the block B is at least $|B|2^{b\ell}R/2^{\ell}$. Thus the probability that m is chosen in a biased column is at most $2^{b\ell}2^{\ell}/2^{b\ell_{max}}\gamma^2|B|R2^{b} < 2^{S}2^{\alpha}2^{\ell}/\gamma^2R2^{b}2^{b\ell_{max}}$. By our choice of ℓ_{max} it will follow that there are at least $|B|2^{b\ell}R/2^{\ell+1}$ nonzero entries in non-biased columns. Hence the probability that B guesses $M(m)$ correctly when m is chosen in a non-biased column is bounded by $1/2 + \frac{\gamma|B|2^{\ell+1}}{|B|R}$.

If follows from the above that

$$Pr[B(m) = M(m)] < 1/2 + 2^{-\alpha} + \frac{2^{S}2^{\alpha}2^{\ell}}{\gamma^2R2^{b}2^{b\ell_{max}}} + \frac{\gamma2^{\ell+1}}{R}$$

Choosing $\alpha = n+2$, $\gamma = 2^{-(n+\ell+3)}$, $\ell_{max} = \frac{S_{max}+4n+3\ell+10}{b}$, we obtain that $Pr[B(m) = M(m)] < 1/2 + 2^{-n}$. Observe that this last choice of ℓ_{max} can be made, despite the fact that ℓ is super linear in ℓ_{max}. Recall that ℓ_{min} is some universal constant, as derived from our protocol, and that $\ell = O(n\ell_{max}\log\ell_{max})$. Recall also that $S_{min} > n$, and that $b \simeq S_{min}/\ell_{min}$. For $S_{max} = n^c$ (where c is some constant), a good choice for ℓ_{max} can always be found, provided that $S_{min} >> cn\log n$.

Acknowledgments: We thank Alfredo De-Santis for sending us the full version of the [7] paper, and Joe Kilian for sending us the full version of the [17] paper.

References

1. S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, "Proof Verification and Hardness of Approximation Problems", 33rd *FOCS, 1992, 14-23.*
2. D. Barrington, "Bounded width polynomial size branching programs recognize exactly those languages in NC^1", 18th *STOC, 1986, 1-5.*
3. M. Blum, A. De Santis, S. Micali, G. Persiano, *Noninteractive Zero-Knowledge,* SIAM Jour. on Computing, Vol. 20, No. 6, December 1991, pp. 1084-1118.
4. G. Brassard, D. Chaum, C. Crepeau, "Minimum disclosure proofs of knowledge", *JCSS, 37, 1988, 156-189.*
5. A. Condon, "Computational models of games", *MIT Press.*
6. B. Chor, O. Goldreich, "Unbiased bits from sources of weak randomness and probabilistic communication complexity", *SIAM J. of Computing, 1988.*
7. A. De-Santis, G. Persiano, M. Yung, "One-message statistical zero-knowledge proofs with space-bounded verifier", 19th *ICALP, 1992.*
8. C. Dwork, L. Stockmeyer, "Finite state verifiers II: zero knowledge" *JACM, 39, 4, 1992, 829-858.*
9. U. Feige, A. Fiat, A. Shamir, "Zero Knowledge Proofs of Identity", *Journal of Cryptology, 1988, Vol. 1, pp. 77-94.*

10. U. Feige, A. Shamir, "Zero Knowledge Proofs of Knowledge in Two Rounds", *Advances in Cryptology – CRYPTO '89 Proceedings, Lecture Notes in Computer Science, Springer-Verlag 435, pp. 526-544.*
11. U. Feige, A. Shamir, *Witness Indistinguishable and Witness Hiding Protocols*, Proc. of 22nd ACM Symposium on Theory of Computing, 1990, pp. 416-426.
12. U. Feige, D. Lapidot, A. Shamir, "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String" *Proc. of* 31st *FOCS, 1990, pp. 308-317.*
13. O. Goldreich, S. Micali, A. Wigderson, *"Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems,* Journal of ACM, Vol. 38, No. 1, July 1991, pp. 691-729.
14. 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.
15. O. Goldreich, H. Krawczyk, "On the composition of zero knowledge proof systems", 17th *ICALP, 1990, 268-282.*
16. J. Kilian, "Uses of randomness in algorithms and protocols", *MIT Press, 1990.*
17. J. Kilian, "Zero knowledge with log space verifiers" 29th *FOCS, 1988, 25-35.*