Interactive Proofs with Provable Security against Honest Verifiers

J. Kilian*

Abstract

Nearly all of the work on constructing zero-knowledge proof systems relies on very strong complexity theoretic assumptions. We consider a form of "no use" zero-knowledge, and show that every language in PSPACE has an interactive proof system that provably achieves "no-use" zero-knowledge against honest verifiers.

1 Introduction.

1.1 Difficulties with proving zero-knowledge.

There are at least two neat things about interactive proof systems [10]. First, they allow one to prove PSPACE hard assertions [19]. Second, they *seem* to allow us to divorce the transfer of confidence from the transfer of knowledge, through the elegant notion of *zero-knowledge interactive proof systems* [10]. Unfortunately, there are few nontrivial languages (e.g., graph nonisomorphism [12]) that are known to have zero-knowledge proof systems. The only way known to show that a language has a zero-knowledge proof system is to show that it has a statistically zero-knowledge proof system. However, such an approach can not give very much generality: Results of Fortnow [7] and Boppana-Hâstad-Zachos [5] imply that if NP has statistically zero-knowledge proofs, then the polynomial-time hierarchy collapses[1]

Such negativism stands in sharp contrast to the bright potential of computational zero-knowledge proof systems. All of IP (and hence all of PSPACE) has zero-knowledge proofs provided that a secure cryptographic bit-committal scheme exists [15]. Secure cryptographic bit-committal schemes can be based on pseudorandom bit generators [17], which in turn can be based on one-way functions [14, 13]. However, we may never be able to prove

*Harvard University and MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139 USA, joek@theory.lcs.mit.edu. Supported by an NSF Postdoctoral Fellowship. Some of the writing up of these results supported by Bell Communications.

[1] See also [1] for more structural theorems about statistical zero-knowledge.

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 378-392, 1991.

the existence of one-way functions. How much security can we actually prove, here and now, without any conjectures? Recent techniques allow us to prove some not completely trivial positive theorems about secure proof systems.

1.2 "No-use" zero-knowledge against honest verifiers.

A massive simplifying assumption one can make is that the verifier obeys the protocol, and only later attempts to obtain extra information. However, even in the honest verifier model, there are no interesting language classes that have zero-knowledge proof systems. Indeed, the negative results of [7, 5, 1] hold for the honest verifier model.

In this paper, we consider a slightly weaker form of security, which is a form of "no-use" zero-knowledge against nonuniform honest verifiers. Various definitions of "no-use" zero-knowledge have been proposed [20, 6]. Our basic intuition is as follows: Seeing a proof that $x \in L$, should not enable one to compute any predicates on x that one could not compute before. This intuition motivates the following definition.

Definition 1 Let (P, V) be an interactive proof system for a language L, and let $(P, V)(x)$ denote the distribution on transcripts produced by running (P, V) on input x. For S an arbitrary subset of L, define $S_k = \{x : x \in S, |x| = k\}$. We say that (P, V) is "no-use" zero-knowledge against honest verifiers if for all $S \subseteq L$, and all predicates Q, the following holds: Suppose there exists a (possibly nonuniform) circuit family $\{C_k\}$ such that $|C_k| \leq k^{c_1}$, and

$$prob(C_k(x, (P, V)(x)) = Q(x)) > \frac{1}{2} + \frac{1}{k^{c_2}},$$

for all $k \in S_k$, where c_1 and c_2 are constants (independent of k). Then there exists a (possibly nonuniform) circuit family $\{C'_k\}$ such that $|C'_k| \leq k^{c_3}$, for some constant c_3 (depending only on c_1, c_2 and L), and $C'_k(x) = Q(x)$ for $x \in S_k$.

The above definitions arguably captures some of our basic intuition about security with respect to an honest verifier. It is certainly far from ideal, but has the vitue of provability. In this extended abstract, we sketch a proof of the following theorem.

Theorem: Any language that has an interactive proof system (P, V) (i.e. any language in PSPACE) has an interactive proof system (P^*, V^*) that is "no-use" zero-knowledge against honest verifiers.

This theorem also allows us to show that other natural notions of security may be achieved. For instance, one may wish to see a proof that $x \in L$, and use this to compute some predicate Q on y, for some other string y. However, for y of length that is some fixed polynomial in x, one can construct a new language L' such that $(x, y) \in L'$ iff $x \in L$, a new predicate $Q'(x, y) = Q(y)$, and apply our theorem.

Our proof uses many fashionable techniques, including

1. Methods for constructing functions that are locally self reducible [2, 16],
2. Shamir's theorem that IP = PSPACE [19],
3. Results on universal hard-core predicates [8], and
4. Zero-knowledge proofs based on ideal bit-commitment [4].

1.3 Outline of the paper

In Section 2, we give the intuition behind our proof. In Section 3, we discuss bit committal based on a PSPACE hard language (QBF for explicitness). In Section 4, we outline the construction of our secure proof system. In Section 5, we outline the argument that the protocol is secure.

2 Overview of the proof.

The essence of our proof is as follows. Suppose one augments the standard model for interactive proof system with an abstract ideal committal scheme, colloquially known as an envelope. Then for any language $L \in IP$ (aka PSPACE), there exists a provably zero-knowledge interactive proof system for L [4]. A reasonable approach is to implement envelopes with a cryptographic protocol that still preserves some of the security properties of abstract envelopes.

A difficulty with this approach is that we must base our commitment scheme on some complexity assumption. However, this assumption will be beyond our ability to prove correct, and so we must allow for the possibility of it being completely wrong! Our approach is to make our committal scheme hard to break relative to the difficulty of simulating the prover. Roughly, if there is a small circuit that can distinguish a committed 0 from a committed 1, then there is a small circuit that can perform the most vital functions of the prover.

How do we make such a committal protocol? We first construct a function F that is computable in PSPACE, and very hard to compute on average. Any circuit that can compute F with nonnegligible probability on a random input of size k^c, for some constant c, can be transformed into a circuit for computing QBF (the set of true quantified boolean formulas) on inputs of size k.[2] We construct such a function by using the techniques of Beaver-Feigenbaum, Lipton, and Yao [2, 16, 21]. We can then use a lemma of Goldreich-Levin [8] to argue that if x and y are chosen at random, then it is hard to compute $b = F(x) \cdot y$, where $F(x)$ and y are treated as boolean vectors. Thus, a random x and y serves to commit a random bit b. This bit may be revealed later by giving an interactive proof that $b = F(x) \cdot y$, using Shamir's protocol [19].

[2] We do not use anything special about QBF - just that it is PSPACE complete.

Given a perfect zero-knowledge proof with envelopes, we now simply replace the envelopes with out commitment scheme. The resulting protocol can be perfectly simulated by a polynomial-time simulator, S^{perf}, that has access to the following two pieces of magic:

1. An oracle that computes QBF on inputs of size $|x|^c$, where c depends on l.
2. An advice distribution that outputs a set of sufficiently many random committals and decommittals for $b = 0$ and $b = 1$, using the appropriate security parameter (based on the size of x).

Unfortunately, this simulation uses too much magic to be immediately useful. Using the simulator for the envelope protocol, we construct a "simulator," S^*, that has access to the advice distribution, but does not have access to the QBF oracle.

We can't prove that the simulation performed by S^* is indistinguishable from the actual protocol. If we are lucky, it will be as good as the real thing insofar as computing predicates is concerned. However, if this is not the case, we can construct a circuit that, using only the advice distribution, can compute QBF on inputs of size $|x|^c$. Using this circuit, we can then perfectly simulate the protocol with just the advice distribution. Finally, we can easily show that in the circuit model of computation access to such an advice distribution does not allow one to compute any new predicates. This is accomplished by a simple "hardwiring" argument, and the fact that the advice distribution depends only on the size of x.

3 Committing bits using our PSPACE hard problem.

In this section, we describe our new bit-committal protocol. The novel feature of our bit is that a small circuit that is able to distinguish a committed 0 from a committed 1 can be transformed into a small circuit to solve bounded sized instances of QBF (quantified boolean formulas). We first show the existence of a function

$$F'_k : \{0,1\}^{k^{c_1}} \rightarrow \{0,1\}^{k^{c_1}},$$

for some constant c_1, such that if for nearly all random x, one can compute $F'_k(x)$ correctly, then one can efficiently decide QBF on problems of size k. Using a construction due Yao [21], we then create a function F_k such that computing F_k with nonnegligible probability on a random input allows one to solve QBF. Finally, we use a result by Goldreich-Levin [8], to create a bit committal scheme with the required properties.

For an additional look at using average-case hard problems to achieve bit-committal (as well as numerous other results in this area), we refer the reader to a forthcoming manuscript by Ostrovsky, Venkatesan, and Yung [18]. Interestingly, they have a completely different way of constructing problem that is PSPACE hard on average, using the more traditional tools of average-case completeness.

3.1 Creating a hard function.

Our construction follows immediately from the methods of Beaver-Feigenbaum and Lipton [2, 16]. Let l be such that $2^l > k+1$, and define the field $\mathcal{F} = GF(2^l)$. Note that $\mathcal{F}$ has at least $k+1$ nonzero elements. Let $QBF_k : \{0,1\}^k \to \{0,1\}$ be the characteristic function of QBF, restricted to k-bit inputs.

First, we define $p_0[x_i] = 1 - x_i$ and $p_1[x_i] = x_i$. Given a boolean k-vector, $\vec{A} = a_1, \ldots, a_k$, we define $\delta_{\vec{A}} : \mathcal{F}^k \to \mathcal{F}^k$ by

$$\delta_{\vec{A}}(x_1, \ldots, x_k) = \prod_{1 \le i \le k} p_{a_i}[x_i].$$

Note that $\delta_{\vec{A}}$ is a multivariate polynomial of degree k. Furthermore, it is not hard to show that for $x_1, \ldots, x_k \in \{0,1\}$, we have

$$\delta_{\vec{A}}(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } x_i = a_i \text{ for } 1 \le i \le k, \text{ or,} \\ 0 & \text{if } x_i \neq a_i \text{ for some } 1 \le i \le k. \end{cases}$$

We define $F'_k : \mathcal{F}^k \to \mathcal{F}$ by

$$F'_k(x_1, \ldots, x_k) = \sum_{\vec{A} \in \{0,1\}^k} QBF_k(\vec{A}) \delta_{\vec{A}}(x_1, \ldots, x_k).$$

Note that F'_k is also a multi-variate polynomial of degree at most k. Furthermore, for $x_1, \ldots, x_k \in \{0,1\}$, we have

$$F'_k(x_1, \ldots, x_k) = QBF_k(x_1, \ldots, x_k).$$

Also, note that F'_k can be uniformly computed in PSPACE.

Now, since F' is a multivariate polynomial of degree k over a field with at least $k+1$ nonzero elements, we can randomly reduce the problem of computing $F'(\vec{X})$ to that of computing

$$F(\vec{Y_0}), \ldots, F(\vec{Y_k}),$$

where each individual Y_i is distributed uniformly over $\mathcal{F}^k$ (implicitly in [2], more explicitly in [3]). Using an argument of Lipton [16], we have the following lemma.

Lemma 1 Suppose there exists a circuit C of size s such that if $x_1, \ldots, x_k \in \mathcal{F}$ is uniformly distributed,

$$prob(C(x_1, \ldots, x_k) = F'_k(x_1, \ldots, x_k)) > 1 - \frac{1}{3k}.$$

Then there exists a circuit C' of size $(sk)^c$ that computes F'_k (and thus QBF_k). Here, c is some global constant. Furthermore, for k sufficently large, F'_k will be nonzero on at least $1/3k$ of its inputs.

Proof: (Sketch) Our proof works the same way as does Lipton's proof of the average case complexity of the permanent. One can compute $QBF_k(x)$ by a locally random reduction to $F'(y_0), \ldots, F'_k(y_k)$, where for each i, y_i is distributed uniformly. Now, for any i, there is at most a $1/3k$ chance that $C(y_i)$ will be different from $F'_k(y_i)$, and thus there is at most a $(k+1)/3k$ chance that $C(y_i)$ deviates from $F'_k(y_i)$ for any $i \in [0,k]$. Hence, if one performs a locally random reduction to from QBF_k to F'_k, and substitutes the output of C' for that of F'_k, one will obtain the correct answer with probability $> \frac{1}{2}$. The probabilistic circuit one obtains can be easily changed into a deterministic circuit with only polynomial blow up.

To see that F'_k must be nonzero with probability at least $1/3k$ (for suitably large k), we note the following fact, which follows straightforwardly from the definition of the Beaver-Feigenbaum reduction: If $F'_k(y_i) = 0$ for all $i \in [0,k]$, then $QBF_k(x)$ must be 0. Since for k sufficently large, there always exists some x such that $QBF_k(x)$ is nonzero, the proof follows from the same argument of Lipton's. ∎

Untuitively, it should be hard to compute F' on a large (but polynomial-sized) set of random inputs, since one of them is bound to be in the hard set. Using an amplification lemma due to Yao, we can create a new function F_k that runs F'_k on several inputs in parallel. Using Yao's technique, and the previous lemma, we can prove the following lemma.

Lemma 2 For $k > 1$, there exists a function, $F_k : k^{c_2} \rightarrow k^{c_2}$, where c_2 is some global constant, with the following property. Suppose there existed a circuit C of size s such that if $x_1, \ldots, x_{k^{c_2}} \in \mathcal{F}$ is uniformly distributed,

$$prob(C(x_1, \ldots, x_{k^{c_2}}) = F'_k(x_1, \ldots, x_{k^{c_2}})) > \epsilon,$$

and $\epsilon > 2^{-k}$. Then there exists a circuit C' of size $(sk/\epsilon)^{c_3}$, where c_3 is some global constant, that computes QBF_k. Furthermore, for k sufficiently large, $F_k(x_1, \ldots, x_{k^{c_2}}$ is nonzero with probability at least $1 - 2^{-k}$. ∎

Clearly, F_k can be computed in PSPACE. It may seem odd that Lemma 2 can allow ϵ to become superpolynomially small. However, in such cases, the bound on the circuit size also becomes superpolynomial.

3.2 Using F_k to commit a bit.

For our protocol, we need to commit and decommit a bit b. In the following protocols, the committor runs in probabilistic PSPACE, and the verifier runs in probabilistic polynomial time.

Protocol COMMIT(b,k) The committor uniformly chooses $X, Y \in \{0,1\}^{k^{c_2}}$ such that $b = F_k(X) \cdot Y$.[3] The committor sends X, Y to the verifier.

Protocol DECOMMIT(b,k,X,Y) The decommittor output b and, using Shamir's protocol, interactively proves to the verifier that $b = F_k(X) \cdot Y$. Shamir's protocol is run many times so as to achieve a maximum error probability of 2^{-k}.

The above protocols form a committal system, if not necessarily a secure one. Once X and Y are announced, $b = F_k(X) \cdot Y$ is completely specified. Thus, the only way a committor can break a decommittal is to give an interactive proof for a false theorem, in which case he will be caught with probability $1 - 2^{-k}$.

We can't show without any assumptions that our bit commitment scheme is secure. However, using the proof of Goldreich-Levin theorem on hard-core bits, and our previous lemmas, we can relate the circuit complexity of breaking our commitment scheme, with security parameter k, and the circuit complexity of computing QBF_k. Crucial to the proof of this lemma was an additional, very powerful proof technique developed at MIT, known as *asking Yishay for help.*

Lemma 3 Let $k > 1$. Suppose there exists a circuit C of size s such that

$$|prob(C(\text{COMMIT}(0,k)) = 1) - prob(C(\text{COMMIT}(1,k)) = 1)| > \epsilon,$$

and $\epsilon > 2^{-k/4}$. Then there exists a circuit C' of size $(sk/\epsilon)^{c_5}$, where c_5 is some global constant, that computes QBF_k. ∎

Proof: (Sketch) We follow the proof of the existence of hardcore predicates for one-way functions [8]. Consider the following problem: Let X be a boolean n-vector, and let $G_X(Y)$ be a function such that

$$prob(G_X(Y) = X \cdot Y) > \frac{1}{2} + \epsilon,$$

for Y chosen uniformly from the set of boolean n-vectors. Let distribution D be equal to $(Y, G_X(Y))$, where Y is uniformly distributed over boolean n-vectors. Given access to D, how well can one guess X? Goldreich and Levin [8] show how, in expected time polynomial in n and $\frac{1}{\epsilon}$, to construct a list of $O(1/\epsilon^2)$ vectors, such that with high probability X is on the list. Thus, one can guess the correct value of X with probability at least $\Omega(\epsilon^2)$.

We now show how to use C to produce a good guessing function $G_{F_k(X)}$ for a nonnegligible fraction of the X's. First, consider the game where b is chosen uniformly, and X and Y are chosen uniformly subject to $b = F_k(X) \cdot Y$. By a simple probability argument it follows that either

$$\begin{aligned} prob(C(X,Y) = F_k(X) \cdot Y) &> \frac{1}{2} + \frac{\epsilon}{2}, \text{ or,} \\ prob(C(X,Y) = F_k(X) \cdot Y) &< \frac{1}{2} - \frac{\epsilon}{2}. \end{aligned}$$

[3] Here, the $\cdot$ operator denotes the dot product.

We now consider the similar case where X and Y are chosen uniformly. Using the fact that $F_k(X)$ is almost always nonzero, we can show that $F_k(X) \cdot Y$ has only a very small bias. Using this fact, and our bound on epsilon, we can then show that either

$$\begin{aligned} prob(C(X,Y) = F_k(X) \cdot Y) &> \frac{1}{2} + \frac{\epsilon}{3}, \text{ or,} \\ prob(C(X,Y) = F_k(X) \cdot Y) &< \frac{1}{2} - \frac{\epsilon}{3}. \end{aligned}$$

Assume without loss of generality that the former case holds (if the latter case holds, simply negate the output of C). Then by a straightforward analysis, with probability at least $\epsilon/2$, a randomly chosen X will have the property that

$$prob(C(X,Y) = F_k(X) \cdot Y) > \frac{1}{2} + \frac{\epsilon}{6},$$

for a uniformly chosen Y. For such X, we can use the Goldreich-Levin algorithm to guess the value of $F_k(X)$ with probability $\Omega(\epsilon^2)$, by using the guessing function,

$$G_{F_k(X)}(Y) = C(X,Y).$$

Even assuming that we fail on all of the other values of X, we can still guess $F_k(x)$ with probability $c\epsilon^3$, for some global constant $c > 0$. This procedure can be hardwired into a circuit of the required size. Finally, for $\epsilon > 2^{-k/4}$, $c\epsilon^3 > 2^{-k}$ for k sufficiently large (constant values of k can be wired in), and we can apply Lemma 2 to give the desired result. ∎

We can use the now standard techniques of Goldwasser-Micali [9] to analyze the security of committing a set of m bits.

Lemma 4 Let $k > 1$, and let B denote some distribution on $\{0,1\}^m$. Given a circuit C (with appropriately many inputs), define $\rho(C,B,k)$ to be the induced distribution on

$$prob(C(\text{COMMIT}(b_1,k),\ldots,\text{COMMIT}(b_m,k)) = 1).$$

Then, if for any $B \in \{0,1\}^m$, there exists a circuit C of size s such that

$$|\rho(C,B,k) - \rho(C,0^m,k)| > \epsilon,$$

and $\epsilon > m2^{-k/4}$. Then there exists a circuit C' of size $(skm/\epsilon)^{c_6}$, where c_6 is some global constant, that computes QBF_k. ∎

4 Converting an interactive proof system into a secure interactive proof system.

In this section, we show that if a proof system (P, V) exists for a language L, then a proof system (P^*, V^*) exists that is "no use" zero-knowledge against honest verifiers. We first review some elementary facts about achieving zero-knowledge with ideal committal schemes (envelopes). We then give our final protocol, along with a "simulator" that will prove crucial to our proof of security.

4.1 Achieving zero-knowledge with envelopes.

Let us review some known results on making protocols one-sided [11] and implementing zero-knowledge with envelopes [4]. Suppose there exists an interactive proof system (P, V) for a language L. If one is given an ideal, information-theoretically secure bit committal protocol (i.e. envelopes), one can convert (P, V) into a new interactive proof system (P', V'), with the following properties.

1. V' runs in probabilistic polynomial time, and P' runs in probabilistic polynomial time, given access to an oracle for $QBF_{|x|^{c_7}}$, for some constant c_7 that depends only on L.
2. For $x \in L$, (P', V') will accept x with probability 1.
3. For $x \notin L$, and all $\hat{P}$, $(\hat{P}, V')$ will accept with probability less than $2^{-|x|}$.
4. For $x \in L$, V''s view of (P', V') is perfectly simulatable in probabilistic polynomial time.

Furthermore, we can construct (P', V') so that a proof proceeds according to the following general format:

Step 1: P' uniformly chooses $b_1, \ldots, b_n \in \{0, 1\}$, where $n = |x|^{c_8}$, and c_8 is a constant depending only on L. P' commits to $b_1, \ldots, b_n$ using the ideal committal system.

Step 2: P' and V' talk back and forth, generating some conversation transcript, T. Both P' and V' are allowed to flip coins during this phase. We can require that V' immediately reveals to P' all of his coin flips (i.e. that the protocol is Arthur-Merlin).

Step 3: P' sends a set $I \subseteq [1, n]$, and for $i \in I$ ideally decommits b_i. V' computes some predicate,

$$accept(x, T, I, \bigcup_{i \in I} b_i).$$

Remark: In our normal form, bits are committed only in the first stage and decommitted only in the third stage. The prover can effectively commit to a new bit, b, in the second

phase by revealing its exclusive-or with a random bit, b_i. Decommitting b_i is then equivalent to decommitting b. Furthermore, the prover can simply state the value of B_i in the second stage, and defer its actual decommittal until Step 3. Note that the verifier has nothing to lose by deferring his abortion of the protocol until Step 3.

We say that a probabilistic circuit S perfectly simulates V''s view of (P', V') if it generates a distribution on

$$(T, I, \bigcup_{i \in I} b_i)$$

that is equal to that induced by (P', V'). Note that S is not required to generate $b_1, \ldots, b_n$ along with T and I. The intuition is that with an ideal commitment scheme, the values of uncommitted bits have nothing to do with V''s view. Note also that S^* implicitly generates V''s coin tosses, since they appear in T.[4]

4.2 Defining our secure protocol.

We now define our secure protocol (P^*, V^*). On input x, (P^*, V^*) execute the following protocol.

Step 1: P^* and V^* start running copies of P' and V'. P' outputs $b_1, \ldots, b_n$. For $1 \leq i \leq n$, P^* runs protocol COMMIT$(b_i, |x|^{c_7})$, generating $(X_1, Y_1), \ldots, (X_n, Y_n)$.

Step 2: P^* and V^* run P' and V' through Step 2 of the original protocol, generating some conversation transcript, T.

Step 3: P' will output a set $I \subseteq [1, n]$. For $i \in I$, P^* and V^* perform protocol DECOMMIT$(b, |x|^{c_7}, X_i, Y_i)$. V^* accepts iff she accepts all of the decommittals and V' accepts on input $(x, T, I, \cup_{i \in I} b_i)$.

4.3 Simulators for our protocol.

The protocol given above is a proof system, since the committal scheme given in Section 3 can only be broken with negligible probability. In order to show that this protocol has some nontrivial security properties, we will construct two simulators. The first simulator gives a perfect simulation, but requires a great deal of outside help. The second simulator requires less help, but gives an imperfect simulation that may be distinguishable from the actual protocol. However, we can show that if one is able to distinguish the second simulation from the actual protocol, then one can implement the first simulation with only a small amount of outside help.

First, we will define our two forms of outside help. The first form of outside help is a $QBF_{|x|^{c_7}}$ oracle. The second form of outside help is an advice distribution $\mathcal{O}(|x|)$. We

[4] In fact, one can produce a simulator $S^*_{\hat{V}}$ that will simulate the (possibly hidden) coin flips of a malicious verifier, $\hat{V}$. However, since we only consider security against an honest verifier, we do not need to worry about such considerations.

define $\mathcal{O}(|x|)$ as the distribution that, for $1 \leq i \leq |x|^{c_8}$ and $b \in \{0,1\}$, outputs (X_i^b, Y_i^b, T_i^b), where

1. X_i^b are Y_i^b are chosen independently from the set of $|x|^{c_2 c_7}$-bit vectors, subject to $b = F_{|x|^{c_7}}(X_i^b) \cdot Y_i^b$, and,
2. T_i^b is chosen from the distribution on transcripts of DECOMMIT$(b, |x|^{c_7}, X_i^b, Y_i^b)$.

Informally, $\mathcal{O}(|x|)$ gives transcripts of the committals and decommittals of $|x|^{c_8}$ 0's and $|x|^{c_8}$ 1's, with security parameter $|x|^{c_7}$.

Our first simulator, denoted $S^{perf}(x)$, proceeds as follows. First, S^{perf} obtains a sample from $\mathcal{O}(|x|)$, then essentially runs the protocol for P^* and V^*. When it is necessary to simulate P', S^{perf} runs the program for P', using the $QBF_{|x|^{c_7}}$ oracle. However, instead of running the protocol for committing bit b_i, S^{perf} simply outputs the values of $(X^{b_i}, Y_i^{b_i})$ provided by $\mathcal{O}$. Similarly, instead of simulating the decommittal of b_i, S^{perf} simply outputs the value of $T_i^{b_i}$ given by $\mathcal{O}$. It is straightforward to verify that $S^{perf}(x)$ gives a perfect simulation for (P^*, V^*) on input $x \in L$.

Our next simulator, $S^*(x)$ works as follows. First, $S^*_{|x|}$ obtains a set of triples (X_i^b, Y_i^b, T_i^b) from $\mathcal{O}(|x|)$. $S^*_{|x|}$ then runs S to obtain $(T, I, \cup_{i \in I} b_i)$. For $i \notin I$, $S^*_{|x|}$ sets $b_i = 0$. $S^*_{|x|}$ then outputs

1. (Step 1) $(X_i^{b_i}, Y_i^{b_i})$, for $1 \leq i \leq |x|^{c_8}$,
2. (Step 2) T, and
3. (Step 3) I, $\bigcup_{i \in I} b_i$, and $\bigcup_{i \in I} T_i^{b_i}$.

Remark: It may seem odd at first that $S^*_{|x|}$ is not necessarily computable by a small probabilistic circuit, but rather needs access to a distribution that may in fact be very hard to generate. However, this doesn't really matter as far as proving no-use zero-knowledge is concerned. Suppose that, when given a sample from some arbitrary but fixed distribution D, a probabilistic circuit C can compute a predicate on a set $\{x_i\}, |x_i| = k$ with an error probability of less than 2^{-k} (taken over D and C's coin tosses). Then we can hardwire C's coin tosses and the output given by D so as to produce a deterministic circuit of the same size that correctly computes the predicate on $\{x_i\}$ with no error.

5 Proving security for our protocol.

We now outline the proof of the main theorem. We first use Lemma 4 to prove the following key lemma.

Lemma 5 Let $x \in L$. Suppose there exists a circuit C of size s such that

$$|prob(C(x,(P^*,V^*)(x)) = 1) - prob(C(x,S^*(x)) = 1)| > \epsilon,$$

where $\epsilon > |x|^{c_8} 2^{-|x|^{c_7}/4}$. Then there exists a circuit of size $(s|x|/\epsilon)^{c_{12}}$ that computes $QBF_{|x|^{c_7}}$. Here, c_{12} is a constant that depends only on L. ∎

Proof: (Sketch) First, we note that S^* correctly outputs many components of the simulation with the correct distribution. By the properties of the simulator for the ideal envelope protocol, it follows that $S^*(x)$ generates the correct distribution for T, I and $\bigcup_{i \in I} b_i$. The definition of $\mathcal{O}$ implies that for $i \in I$, (X_i, Y_i) and T_i will be distributed correctly as well. Let Q represent the set of this correctly simulated material, i.e.,

$$Q = \left(T, I, \bigcup_{i \in I} (b_i, (X_i, Y_i), T_i)\right).$$

We can view $S^*(x)$ and $(P^*,V^*)(x)$ as outputting

$$\left(Q, \bigcup_{i \notin I} (X_i, Y_i)\right),$$

according to (possibly) different distributions. We can further imagine that S^* and (P^*,V^*) first generate Q, and then generate the second component of the output distribution based on Q. This conceptual view is not in accord with the procedures we have specified for generating the distributions in question, but can be justified using conditional probabilities. We denote the distributions induced on the second component by $S^*(x,Q)$ and $(P^*,V^*)(x,Q)$.

When we project onto the first coordinate (looking at the value of Q), the distributions of S^* and $(P^*,V^*)(x)$ are identical. By a straightforward probabilistic argument, it follows that there is some value of Q such that

$$|prob(C(x,(Q,(P^*,V^*)(x,Q))) = 1) - prob(C(x,(Q,S^*(x,Q)) = 1))| > \epsilon.$$

We can hardwire Q into C to obtain a circuit C' that distinguishes $(P^*,V^*)(x,Q)$ from $S^*(x,Q)$. To complete the proof, is suffices to show that a circuit that distinguishing between these two distributions on $\bigcup_{i \notin I}(X_i,Y_i)$ can be transformed into a small circuit for $QBF_{|x|^{c_7}}$.

Given x and Q, $(P^*,V^*)(x,Q)$ can be generated by choosing b_i, for $i \notin I$, according to some (conditional) distribution, and then choosing (X_i,Y_i) according to b_i and the bit committal procedure.[5] $S^*(x,Q)$ can be generated by choosing $b_i = 0$ for $i \notin I$ and then

[5]This is one of the points in our proof where the honesty of the verifier is required. We implicitly assume that V^* does not look at the X_i, Y_i vectors; otherwise, this last claim would be false.

choosing (X_i, Y_i) according to b_i and the bit committal procedure. The only difference is in the distribution on the b_i's (for $i \notin I$). Thus, we can let B be the distribution on b_i, $i \notin I$, generated by $(P^*, V^*)(x)$, and apply Lemma 4 to complete the proof. ∎

Using Lemma 5, it is relatively straightforward to complete the proof of our main theorem.

Theorem 1 Let (P^*, V^*) be the interactive proof system for L described above. Then (P^*, V^*) is no-use zero-knowledge against honest verifiers.

Proof: (Sketch) Suppose there existed a set $S \subseteq L$, a predicate Q, a constant c, and a circuit family $\{C_k\}$, such that for $x \in S$,

$$prob(C_{|x|}(x, (P^*, V^*)(x)) = Q(x)) > \frac{1}{2} + \frac{1}{|x|^c}.$$

First, we only consider the case where $1/4|x|^c > |x|^{c_8} 2^{-|x|^{c_7}/3}$. For any value of c, the set of possible x that violate this constraint is finite, and can thus be handled by a look-up table.

We construct a circuit family $\{C'_k\}$ for Q as follows. For each k, we consider the probabilistic circuit $C_k^{sim}(x)$, that has access to a random output of $\mathcal{O}(k)$. $C_k^{sim}(x)$ simply runs $C_k(x, S_k^*(x))$. Now, one of the following cases must hold,

1. For all $x \in S, |x| = k$, we have

$$prob(C_k^{sim}(x) = Q(x)) > \frac{1}{2} + \frac{1}{2k^c}, \text{ or,}$$

2. For some $x_0 \in S, |x_0| = k$, we have

$$|prob(C_k(x_0, (P^*, V^*)(x_0)) = 1) - prob(C_k(x_0, S_k^*(x_0)) = 1)| > \frac{1}{4k^c}.$$

In the first case, we can perform standard amplification on $C_k^{sim}(x)$ to construct a circuit $C'_k(x)$ of size $|x|^{c_{10}}$, where c_{10} is a constant depending only on L and c, such that,

1. $C'_k(x)$ has access to (perhaps many) outputs from $\mathcal{O}(k)$.
2. For all $x \in S, |x| = k$, we have,

$$prob(C'_k(x) \neq Q(x)) < 2^{-k}.$$

We can then hardwire the random and auxiliary inputs (from $\mathcal{O}$) of C', so as to ensure that $C'_k(x) = Q(x)$ for all $x \in S, |x| = k$.

In the second case, we can use Lemma 5 to produce a circuit for $QBF_{k^{c_7}}$ of size $k^{c_{11}}$, where c_{11} is a constant depending only on L and c. Using these small circuits, we can then construct a circuit that simulates S_k^{perf} (S^{perf} restricted to inputs of size k), which is of size $k^{c_{12}}$ (where c_{12} is a constant depending only on L and c), and which only uses an output from $\mathcal{O}(k)$ (since all of the calls to the $QBF_{k^{c_7}}$ oracle are replaced by the actual small circuit). We can then construct a circuit $C_k^*(x)$ that simply runs $C_k(x, S_k^{perf}(x))$, which is in fact equivalent to running $C_k(x, (P^*, V^*)(x))$. We have, for all $x \in S$, $|x| = k$,

$$prob(C_k^*(x) = Q(x)) > \frac{1}{2} + \frac{1}{k^c}.$$

Then, by the same construction by which we generated a small C'_k from C_k^{sim}, we can construct a circuit C'_k from C_k^*, of size $k^{c_{13}}$, where c_{13} is a constant depending only on c and L. ∎

6 Acknowledgments.

I would like to acknowledge substantial technical help from Yishay Mansour, and some very useful comments on our definition of security by Shafi Goldwasser.

References

[1] B. Aeillo and J. Håstad. Perfect zero-knowledge languages can be recognized in two rounds. Proc. FOCS87.

[2] D. Beaver and J. Feigenbaum. Hiding Instances in Multioracle Queries, *Proc. of the 7th STACS* (1990), Springer Verlag LNCS 415, 37–48.

[3] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Cryptographic Applications of Locally Random Reductions, AT&T Bell Laboratories Technical Memorandum, November 15, 1989.

[4] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali, and P. Rogaway. Everything Provable is Provable in Zero-Knowledge, Proc. of CRYPTO88.

[5] R. Boppana, J. Håstad, and S. Zachos. Does CoNP Have Short Interactive Proofs? IPL, 25, May 1987, P'. 127-132.

[6] C. Dwork and L. Stockmeyer, Interactive Proof Systems with Finite-State Verifiers, Proc. CRYPTO88.

[7] L. Fortnow. The Complexity of Perfect Zero-Knowledge. Proc. STOC87.

[8] O. Goldreich and L. Levin, Hardcore predicates from all one-way functions. Proc. STOC89

[9] S. Goldwasser and S. Micali. Probabilistic Encryption, *J. Comput. System Sci.* **28** (1984), 270–299.

[10] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems, *SIAM J. Comput.* **18** (1989), 186–208.

[11] O. Goldreich, Y. Mansour, and M. Sipser. Interactive Proof Systems: Provers that never fail and random selection. Proc. FOCS87.

[12] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design, Proc. of FOCS86.

[13] Johan Håstad. Manuscript.

[14] R. Impagliazzo, L. Levin, and M. Luby. Pseudo-random Generation from One-Way Fynctions, Proc. STOC89

[15] R. Impagliazzo and M. Yung. Direct Minimum-Knowledge Computation, Proc. of CRYPTO87.

[16] R. Lipton. New Directions in Testing, manuscript, October, 1989.

[17] M. Naor Pseudorandomnesss and Bit Commitment, Proc. of Crypto89.

[18] R. Ostrovsky, Venkatesan and M. Yung. Manuscript (Submitted to STOC91)

[19] A. Shamir. IP = PSPACE, manuscript, December 26, 1989.

[20] A. Shamir. CRYPTO Rump Session.

[21] A. C. Yao. Protocols for Secure Computations, Proc. of FOCS82.