Perfect Zero-Knowledge Languages can be Recognized in Two Rounds

William Aiello* Johan Hastad**

Laboratory of Computer Science
Department of Mathematics
Massachusetts Institute of Technology

Abstract: A hierarchy of probabilistic complexity classes generalizing NP has recently emerged in the work of [Ba], [GMR], and [GS]. The IP hierarchy is defined through the notion of an interactive proof system, in which an all powerful prover tries to convince a probabilistic polynomial time verifier that a string w is in a language L. The verifier tosses coins and exchanges messages back and forth with the prover before he decides whether to accept w. This proof-system yields "probabilistic" proofs: the verifier may erroneously accept or reject w with small probability.

In [GMR] such a protocol was defined to be a *zero-knowledge protocol* if at the end of the interaction the verifier has learned nothing except that $w \in L$. We study complexity theoretic implications of a language having this property. In particular we prove that if L admits a zero-knowledge proof then L can also be recognized by a two round interactive proof. This complements a result by Fortnow [F] where it is proved that the complement of L has a two round interactive proof protocol.

The methods of proof are quite similar to those of Fortnow [F]. As in his case the proof works under the assumption that the original protocol is only zero-knowledge with respect to a specific verifier.

1. Introduction

Interactive proofs were independently introduced by Babai [Ba] and Goldwasser, Micali and Rackoff [GMR]. Informally, an interactive proof can be viewed as a game between an infinitely powerful prover and a verifier restricted to random polynomial time. The prover tries to convince the verifier of a fact like $w \in L$ for some specific input w where L is a given language. We say that a language L has an interactive proof system if for every $w \in L$, the prover can convince the verifier of this with probability at least $1 - 2^{-|w|}$ and if for every $w \notin L$ no prover can convince the verifier that $w \in L$ with probability greater than $2^{-|w|}$.

The fact that the verifier is allowed to be probabilistic and that there is a small probability of error is essential in the definition. If these two features were not allowed, only languages in NP would have interactive proofs. With the present definitions the class contains graph non-isomorphism [GMW] which is not known to be in NP.

Babai [Ba] considered interactive proofs for complexity theoretic purposes while Goldwasser, Micali and Rackoff's [GMR] primary motivation was cryptography. Babai proved that any language that could be recognized in a constant number of interactions could be recognized in two interactions. The class of languages that can be recognized in two rounds will be denoted by $IP[2]$ or simply IP. In general, a language that can be recognized in $Q(|w|)$ rounds is said to belong to $IP[Q]$. Babai conjectured that $IP[Q] = IP[2]$. This conjecture is still open but it has been shown to be false in a relativized setting. Aiello, Goldwasser and Hastad [AGH] proved that for any pair of polytime constructible functions f and g such that f/g is unbounded there is an oracle B such that $IP^B[f] \not\subseteq IP^B[g]$. On the positive side Moran [M] proved that for any constant c, $IP[Q] = IP[cQ]$.

Goldwasser, Micali and Rackoff defined a more general model of interactive proof than did Babai. However, Goldwasser and Sipser [GS]

* Supported by an ONR fellowship and partially supported by NSF grant DCR-8509905.

** Supported by an IBM Post Doctoral Fellowship and partially supported by Air Force Contract AFOSR-86-0078.

subsequently showed that the two models are equivalent from a complexity point of view. We will here use only to the GMR model. In addition to defining interactive proofs Goldwasser, Micali, and Rackoff defined zero-knowledge protocols. Intuitively, a zero-knowledge interactive proof system for a language L has the property that it convinces the verifier that $w \in L$ but gives him no additional knowledge. No additional knowledge can be interpreted in three ways:

(1) No knowledge in an "information theoretic" sense.

(2) Almost no knowledge in an "information theoretic" sense.

(3) No knowledge which can be extracted in random polynomial time.

In this paper we will be concerned with the first two kinds which will be called perfect and statistical zero-knowledge respectively. The third type is usually referred to as computational zero-knowledge.

Recently Fortnow [F] discovered some strong complexity theoretic implications of the zero-knowledge concept. He showed that if L admits a perfect or statistical zero-knowledge proof system of an arbitrary number of rounds then the complement, $\bar{L}$, is contained in $IP[2]$. Our main result is to extend his argument to show that L is also contained in $IP[2]$. Hence, under the assumption that $IP[poly] \neq IP[2]$, the zero-knowledge constraint severely restricts the power of many-round interactive proofs. The results in this paper do not depend on any unproven cryptographic assumptions.

The outline of the paper is as follows: In section 2 we give the definitions and also state the main theorem. We give the intuition behind the proof in section 3. In section 4 we recall some facts about estimating sizes of sets using interactive proofs and finally in section 5 we give a fairly complete proof of the theorem.

2. Notation and Definitions

In this section we give the formal definitions needed for the paper. The prover will be denoted by P and the verifier by V. As mentioned in the introduction, P is all powerful while V is a probabilistic polynomial time Turing machine. On input w, P and V interact in rounds in the following way.

(1) V makes a random polynomial time computation based on the conversation so far, the input w and the contents of his memory.

(2) V transmits the result of the computation to P. We will denote the message sent by V in round i by x_{2i-1}.

(3) P makes an arbitrary computation based on the conversation so far and the input.

(4) P transmits the result to V. We will denote the message sent by P to V in round i by y_{2i}.

The interaction is terminated by the verifier accepting or rejecting the input.

Let $P \leftrightarrow V(w)$ denote a transcript of the interaction between the prover and the verifier. This is of course a stochastic variable depending on P's and V's random choices. P and V form an interactive proof system for a language L iff

(1) If $w \in L$ then $\Pr[P \leftrightarrow V(w)$ makes V accept$]$ $> 1 - 2^{-|w|}$ for any w.

(2) If $w \notin L$ then even with an optimal prover $\Pr[P \leftrightarrow V(w)$ makes V accept$] < 2^{-|w|}$ for any w.

Definition: A language L which has an interactive proof system where the number of interactions is bounded by $Q(|w|)$ on input w is said to belong to the class $IP[Q]$.

The error probability, i.e., the maximum probability that the verifier erroneously accepts or rejects, may vary in the definition of an interactive proof system. Sometimes this probability is chosen to be $1/3$, sometimes $2^{-|w|}$. However, it is not hard to see that the two definitions are equivalent by running the protocol several times and taking the majority of the outcomes.

2.1 Zero-Knowledge

In this section we will give the formal definition of a language having a zero-knowledge interactive proof system. We will first need some properties of probability distributions on strings.

Given two parametrized probability distributions $A[w]$ and $B[w]$ and let $Pr_{A[w]}(y)$ and $Pr_{B[W]}(y)$ denote the probability of y according to the distributions respectively. We use the following conventions.

(1) $A[w] = B[w]$ if $Pr_{A[w]}(y) = Pr_{B[w]}(y)$ for all y and all w.

(2) $A[w]$ is ***statistically close*** to $B[w]$ if $\sum_y |Pr_{A[w]}(y) - Pr_{B[w]}(y)| \leq 1/|w|^c$ for any c and sufficiently long w.

(3) $A[w]$ is ***polytime indistinguishable*** from $B[w]$ if for any random polynomial time computable predicate P

$$|\sum_{y \in P^{-1}(1)} Pr_{A[w]}(y) - Pr_{B[w]}(y)| < \frac{1}{|w|^c}$$

for any c and sufficiently long w.

The key to the formal definition of zero-knowledge is that of a ***simulator.*** A simulator M_V is a random expected polynomial time machine that on input w produces strings with a probability distribution which is close to the distribution of strings produced by P and V interacting. Let us write this down as a formal definition. Let $P \leftrightarrow V[w]$ be the distribution corresponding to the random variable $P \leftrightarrow V(w)$ and let $M_V[w]$ be the distribution that M_V produces on input w. Corresponding to the the three definitions above there are three types of zero-knowledge. An IP protocol accepting L is

(1) ***perfect zero-knowledge*** if for any verifier V' there exists a simulator $M_{V'}$ such that for all $w \in L$, $P \leftrightarrow V'[w] = M_{V'}[w]$.

(2) ***stastistical zero-knowledge*** if for any verifier V' there exists a simulator $M_{V'}$ such that for all $w \in L$, $P \leftrightarrow V'[w]$ is statistically close to $M_{V'}[w]$.

(3) ***computational zero-knowledge*** if for any verifier V' there exists a simulator $M_{V'}$ such that for all $w \in L$, $P \leftrightarrow V'[w]$ is polytime indistinguishable from $M_{V'}[w]$.

For all three of the above definitions there is also a less restrictive notion. Given an interactive proof between P and V which recognizes a language, L, the protocol is ***zero-knowledge for a trusted verifier*** if there exists a single simulator M such that for $w \in L$, $M[w]$ is close to $P \leftrightarrow V[w]$ in one of the above senses. Clearly, any protocol which is zero-knowledge is also zero-knowledge for a trusted verifier.

Fortnow [F] proved that if L admits an arbitrary round proof which is perfect or statistical zero-knowledge for a trusted verifier then the complement of L is in $IP[2]$. Our main result is:

Theorem: *Suppose that L admits an arbitrary round proof which is perfect or statistical zero-knowledge for a trusted verifier, then $L \in IP[2]$.*

We postpone the proof of the theorem to section 5. In the next section we give the intuition of the proof.

Observe that it is probably too much to hope that these results would extend to computational zero-knowledge by a similar proof. The reason being that any language that has an interactive proof has a computational zero-knowledge proof ([GMW], [Be]) provided that some reasonable cryptographic assumptions are true. Thus a similar theorem for the case of computational zero-knowledge would give a proof that $IP[Q] = IP[2]$ for any Q. Such a proof could not relativize to an arbitrary oracle since there are oracles such that this is false [AGH]. Thus a proof like ours could not work since our proof relativizes.

3. Outline of Proof; Structure of Simulator

For the time being let us consider protocols which are ***perfect*** zero-knowledge for a trusted verifier. Let L be recognized by such a protocol between P and V and let the simulator be M. Our goal is to show that L can also be recognized by a bounded round interactive proof. Alice, or A, will be the prover in the bounded round proof and Bob, or B, will be the verifier. So far we have two prover-verifier pairs but let us introduce one more. In conversations produced by the simulator we will say that the prover-moves in the conversation are produced by a ***virtual*** prover, P', and the verifier-moves are produced by a *virtual* verifier, V'. A very broad outline of the Alice-Bob protocol is as follows. On input w Alice will try to convince Bob of two things:

(1) $M(w)$ accepts with probability greater than $2/3$.

(2) P' plays *honestly.*

A move by P' is honest if it depends only on the conversation so far. This will be clarified below. Bob will accept w only if he is convinced of both (1) and (2).

Assume for the time being that $M(w)$ accepts with probability greater than $2/3$. If not Bob will correctly reject with high probability. Let us examine the difference between P''s play when $w \notin L$ and $w \in L$. When $w \notin L$ then any real prover can only win the game with probability at most $2^{-|w|}$. But by assumption P' wins the game with probability $> 2/3$. This difference is possible since the virtual prover "knows" the coins of the simulator and hence the virtual verifier. He can make moves targeted towards these coins while any real prover only has the information given by the conversation so far. Furthermore, the virtual prover *has* to use this advantage to get the high probability of winning. Alice will not be able to convince Bob that all of P''s moves are honest and Bob will correctly reject with high probability.

When w is in the language the virtual prover behaves exactly as the real prover by definition of perfect zero knowledge. Thus the virtual prover's moves depends only on the conversation so far. He plays honestly everywhere and Bob will properly accept with high probability.

Let us make the definition of honest more formal. Let r be the V's coins, $|r| = l(n)$ where $l(n)$ is a polynomial. Let x be the V's moves and y be the P's moves. Let s_k be a string produced by the first k interactions of the $P \leftrightarrow V$ protocol, $s_k = x_1 y_2 \ldots q_k$ where $q_k = x_k$ for k odd or y_k for k even. If $d(n)$ is the number of interactions then we will write $s_{d(n)}$ as s. Without loss of generality V will send his coins to P on the very last move. So, the entire dialogue is $P \leftrightarrow V(w) = s, r$. Interactions which start with s_k and end with r are written $P \leftrightarrow V(w) = s_k *, r$. Recall that the verifier's $j+1$st move is a function of the input, its random bits and the previous $2j$ interactions. We write this as $V(w, r, s_{2j}) = x_{2j+1}$. Let α_{s_k} be the set of all the verifier's coins that are *consistent* with the conversation s_k. That is, α_{s_k} is the set of all r such that $V(w, r, s_{2i}) = x_{2i+1}$ for $0 \leq 2i < k$. This implies in particular that for k odd $\alpha_{s_k} = \alpha_{s_k y}$ for all y.

Obviously the conditional probability that V outputs r on his last move given that the conversation thus far is s_k is the same for all r consistent with s_k:

$$Pr[P \leftrightarrow V(w) = s_k *, r \,|\, P \leftrightarrow V(w) = s_k] = \frac{1}{|\alpha_{s_k}|}.$$

This is true for all k and s_k. If k is odd then $\alpha_{s_k} = \alpha_{s_k y}$ and hence

$$Pr[P \leftrightarrow V(w) = s_k y *, r \,|\, P \leftrightarrow V(w) = s_k y] = \frac{1}{|\alpha_{s_k}|}.$$

Observe that the above implies that for all moves y of the prover

$$Pr[P \leftrightarrow V(w) = s_k y *, r \,|\, P \leftrightarrow V(w) = s_k *, r] =$$

$$Pr[P \leftrightarrow V(w) = s_k y \,|\, P \leftrightarrow V(w) = s_k]$$

for all r. A move by the virtual prover which has the same property is an *honest* move. That is, y is an honest move if

$$Pr[P' \leftrightarrow V'(w) = s_k y *, r \,|\, P' \leftrightarrow V'(w) = s_k *, r] =$$

$$Pr[P' \leftrightarrow V'(w) = s_k y \,|\, P' \leftrightarrow V'(w) = s_k]$$

for all r. In words, a move y is honest if for all r the probability that P' plays y given that s_k has been played so far and V' has coins r is equal to the probability that P''s move is y given that the conversation thus far is s_k. That is, P''s move is based only on the conversation so far and not upon additional information about r. Since $Pr[P' \leftrightarrow V' = s, r] = Pr[P \leftrightarrow V = s, r]$ for all s, r when $w \in L$, an immediate consequence of the definition is that all of P''s moves are honest when $w \in L$.

P' cheats on move y if it does use additional information about r. That is, y is a cheating move if for some r the above equality does not hold. It cheats badly if the relation is far from equality. When $w \notin L$ we will prove that P' cheats badly on a significant fraction of its moves in order to win the game with high probability. This will be made precise in Lemmas 5.3 and 5.4.

The goal of the Alice and Bob protocol is to distinguish between P' playing honestly everywhere and P' cheating a great deal. With Alice's help Bob will recognize when P' is honest and will accept. However, Alice will not be able to fool Bob into accepting very often when P' cheats a great deal.

Since our A-B protocol will be composed of subprotocols which deal with the size of sets let us reformulate the definition of an honest move. Let c be the simulator's coin, $|c| = q(n)$ where $q(n)$ is a polynomial. Let β_{s_k} and $\beta_{s_k*,r}$ be the sets of c for which $M(w) = s_k$ or s_k*, r respectively. Now, P''s move y is honest when

$$\frac{|\beta_{s_k y*,r}|}{|\beta_{s_k*,r}|} = \frac{|\beta_{s_k y}|}{|\beta_{s_k}|}$$

for all r.

The following section deals with the aforementioned subprotocols which prove upper and lower bounds on the size of sets.

4. Upper and Lower Bound Protocols

Let us first consider Babai's [Ba] subprotocol for proving lower bounds on the size of sets. His analysis is based on a lemma of Sipser's [S]. Suppose $C \subseteq \Sigma^k$ where membership in C is testable in polynomial time. Let H be a $k \times b$ Boolean matrix and let $h:\Sigma^k \to \Sigma^b$ be defined by matrix multiplication modulo 2, $h(x) = xH$. The protocol "P proves $|C| \geq 2^b$" is as follows:

1. P sends b to V.
2. V picks a random $k \times b$ matrix H and a random element z of Σ^b. V sends H and z to P.
3. P responds with $c \in \Sigma^k$.
4. V accepts iff $c \in C$ and $h(c) = z$.

Lemma 4.1:

(1) $Pr[V \text{ accepts}] \geq 1 - \frac{2^b}{|C|}$.

(2) $Pr[V \text{ accepts}] \leq \frac{|C|}{2^b}$.

Proof: Proof of (2) is straightforward. Proof of (1) follows from the fact that the events $h(c) = z$ and $h(c') = z$ are pairwise independent. Hence, we can use Chebychev's inequality. Let X be the number of elements $c \in C$ for which $f(c) = z$. Note that $\mu(X) = |C|/2^b$ and $\sigma^2 \leq |C|/2^b$.

$$Pr[V \ rejects] = Pr[X = 0] \leq$$

$$Pr[|X - \mu| \geq |C|/2^b] \leq 2^b/|C|. \quad \square$$

We will need an extension of this protocol. In many cases we will have a set C and to every $c \in C$ there will be a set D_c. Membership in C and each D_c can be tested in polynomial time. We will need to prove a statement like: "There are at least 2^a c's in C such that $|D_c| \geq 2^b$." Assume that $C \subset \Sigma^k$ and $D_c \subset \Sigma^l$ for all c. The protocol will be the following:

1. P sends a and b to V.
2. V picks a random $k \times a$ matrix H_1 and a random element z_1 of Σ^a. V sends H_1 and z_1 to P.
3. P responds with $c \in \Sigma^k$.
4. V checks that $c \in C$ and $h_1(c) = z_1$. If not he rejects. V picks a random $l \times b$ matrix H_2 and a random element z_2 of Σ^b. V sends H_2 and z_2 to P.
5. P responds with $d \in \Sigma^l$.
6. V accepts iff $d \in D_c$ and $h_2(d) = z_2$.

There are many possibilities for the distribution of the sizes of D_c but we will analyze only the cases we need for the main theorem.

Lemma 4.2:

(1) *If there are at least $r2^a$ $c \in C$ such that $|D_c| \geq r2^b$ then the probability that V accepts is at least $1 - \frac{2}{r}$.*

(2) *If there are at most $r2^a$ $c \in C$ such that $|D_c| \geq r2^b$ then the probability that V accepts is at most $2r$.*

Proof: Lemma 4.2 follows from using Lemma 4.1 twice. $\square$

Next we present a protocol developed by Fortnow [F] for proving upper bounds on the size of sets. Suppose the verifier has a random element, c, of the set $C \subseteq \Sigma^k$. Define "P proves $|C| \leq 2^b$" as follows.

1. P sends b to V.
2. V picks a random $k \times b$ matrix H and calculates $h(c) = z$. V sends H and z to P.
3. P sends a to V.
4. V accepts iff $a = c$.

Lemma 4.3:
(1) $Pr[V \text{ accepts }] \geq 1 - \frac{|C|-1}{2^b}$.
(2) $Pr[V \text{ accepts }] \leq \frac{d2^b}{|C|-1}$ where $d = 3 + \sqrt{5}$.

Proof: Proof of (1) is straightforward. Proof of (2) follows. Let S be the number of elements of C which map to z.

$$Pr[V\ accepts] \leq \Sigma_{j=1}^{|C|} \frac{1}{j} Pr[S = j] \leq Pr[S \leq i] + \frac{1}{i}$$

for all i. Again we will apply Chebychev's inequality. Note that $\mu(S) = 1 + \frac{|C|-1}{2^b}$ and $\sigma^2 \leq \mu$. So, for $j < \mu$

$$Pr[S \leq j] \leq Pr[\mu - S \geq \mu - j] \leq \frac{\mu}{(\mu - j)^2}.$$

For $j = \frac{2}{3+\sqrt{5}}\mu$ we get

$$Pr[V\ accepts] \leq \frac{(3 + \sqrt{5})2^b}{|C| - 1}. \quad \square$$

5. Proof of Main Theorem

In this section we give a fairly complete outline of the formal counterpart of the argument outlined in section 3. We assume that the protocol is perfect zero-knowledge and that the simulator is polynomial time instead of expected polynomial time. The proof goes through without change (except for more cumbersome notation) for statistical zero-knowledge with respect to a polynomial time simulator.

When the simulator is expected polynomial time one has to be more careful. In all the known perfect and statistical zero-knowledge protocols there is a polynomial $p(n)$ such that the simulator halts within time $p(n)$ with exponentially high probability. It is easy to convert such a simulator to a simulator that runs in polynomial time and gives a statistical zero-knowledge simulation. However, the definitions allow the simulator to run, for example, for time $p(n)$ with probability $t(n)/p^2(n)$ where $t(n)$ is some polynomial. In this case our protocol needs slight modification and the proof gets more complicated. We will include the necessary details in the full paper.

Suppose that the original protocol contained $d(n)$ interactions when applied to an input w with $|w| = n$. Without loss of generality we will assume that $d(n) > n$ since fewer interactions will only help us. We will also assume that n is sufficiently large which implies that any growing function of n will be assumed to be larger than any constant and any function exponential in n will be larger than a function which is polynomial in n. Furthermore we will assume that the error probability is bounded by $2^{-2nd(n)}$. By this we mean that if $w \in L$ then the prover can convince the verifier with probability $1 - 2^{-2nd(n)}$ and if $w \notin L$ then the probability that an optimal prover can convince the verifier is at most $2^{-2nd(n)}$. If these conditions are not met for the original protocol, then they can be achieved by playing the original protocol in parallel and taking the majority of the outcomes. The original protocol is perfect zero-knowledge for a trusted verifier and remains so when run in parallel. Note that this is not necessarily the case for zero-knowledge protocols.

Before we formally define the protocol let us dispose of the easy cases. We will assume that the simulator almost always makes the verifier accept. For the inputs w for which this is not the case it is easy to detect that $w \notin L$. Indeed if B ever sees a conversation generated by the simulator where the verifier rejects he should also reject the input. Using the same argument we can assume that if the virtual verifier reveals coins r in the end, then its previous moves are almost always the moves that the real verifier would have made provided it had coins r.

Remember that in simulating the original protocol, M only uses $q(n)$ coins and hence the probability of any event concerning the simulation is a multiple of $2^{-q(n)}$.

The Alice-Bob protocol will consist of two parts. The first is rather trivial part and checks that the simulator actually outputs conversations for most r. The second will refer to a protocol which is the original P-V protocol run $nd^2(n)q^2(n)$ times independently in parallel. Thus each message in this protocol consists of $nd^2(n)q^2(n)$ parts regarding different independent conversations. We denote the simulator for this game by M_1, the virtual verifier by V_1, and the virtual prover by P_1. Thus M_1 just runs $nd^2(n)q^2(n)$ independent copies of M. We will denote the supermessages produced by M_1 by capital letters. Let C^M be the coins of M_1, $C^M = c^{(1)}c^{(2)}\dots c^{(nd^2(n)q^2(n))}$, and R^V be the coins of the verifier of the parallel P-V protocol, $R^V = r^{(1)}r^{(2)}\dots r^{(nd^2(n)q^2(n))}$. Let us define the A-B protocol.

Protocol A-B

0. Bob picks a random r and Alice proves that $|\beta_{*,r}| \geq 2^{q(n)-l(n)-2}$. In fact run this protocol 100 times in parallel and Bob accepts if the majority of the protocols lead to accept.

For $j = 1, 3, 5, \dots d(n)$ in parallel and n times in parallel for each j do:

1. Bob chooses a random seed C^M for M_1, runs M_1 and sends $S_j = X_1Y_2\dots X_j$ to Alice.
2. Alice responds with numbers a_j, b_j and c_j satisfying $a_j - (b_j + c_j) \leq 15$.
3. Alice proves that $|\beta_{S_j}| < 2^{a_j}$ and that there are at least 2^{b_j} R^V's which are consistent with S_j and such that $|\beta_{S_j*,R^V}| \geq 2^{c_j}$.
4. Alice sends Bob a number d_j.
5. Alice proves that there are at least 2^{d_j} moves Y_j by P_1 such that for each of these moves there are at least 2^{b_j} R^V's for V_1 consistent with S_j such that $|\beta_{S_jY_j*,R^V}| > 2^{c_j-d_j}/nd^2(n)q^3(n)$.
6. Bob accepts iff he for every j accepts in a majority of the subprotocols regarding conversations of length j and all the conversations he has seen produced by the simulator have led to the verifier accepting.

As stated the protocol contains more than two rounds. However, by Babai's result any language that can be recognized in a constant number of rounds can also be recognized in two rounds. We have to prove that Alice can convince Bob precisely when $w \in L$, i.e., we need to establish the following two lemmas.

Lemma 5.1: *If $w \in L$ Alice can convince Bob with probability $\frac{9}{10}$.*

Lemma 5.2: *If Alice can convince Bob with probability $\frac{1}{10}$ then $w \in L$.*

Lemma 1 is quite easy to prove and we omit the proof. Lemma 2 will require a little bit of work. We will assume that $w \notin L$ and that Alice can convince Bob with probability 1/10 and obtain a contradiction. We start by analyzing how M, the simulator for the original protocol, behaves.

Recall the definition of honest from section 3. For shorthand let $P_{s_k,r}(y) = Pr[M = s_ky*, r|M = s_k*, r]$ and $P_{s_k}(y) = Pr[M = s_ky|M = s_k]$. For any initial conversation s_k we will say that the virtual prover *cheats* on move y and coin r if $P_{s_k,r}(y) > 2^n P_{s_k}(y)$. One consequence of the definition is that for a fixed y the probability that y is a cheating move for a random r is at most 2^{-n}.

Look at the tree of all conversations output by M which end with the virtual verifier revealing coins r. Let the conditional probability $w_{s,r} = Pr[M = s, r|M = *, r]$ be the weight of the leaf associated with conversation s. There is a set S of conversations where the virtual verifier accepts and the virtual prover does not cheat. Let k_r be the total weight of these conversations:$k_r = \sum_{s\in S} w_{s,r}$.

Lemma 5.3: $\sum_r k_r \leq 2^{l(n)}2^{-\frac{3}{2}nd(n)}$

Proof: Define a new prover P^{new} which for all initial conversations s_k plays y with probability $P_{s_k}(y)$. Consider the game where P^{new} plays against the real verifier, V, and V has coins r.

Let the conditional probability $w'_{s,r} = Pr[P^{new} \leftrightarrow V = s, r | P^{new} \leftrightarrow V = *, r]$ be the weight of the leaves associated with the game tree. The set S naturally corresponds to a set of leaves. Since the virtual prover did not cheat we know that the weight of any such leaf can only go down by a factor $2^{-nd(n)/2}$, i.e., $w'_{s,r} \geq 2^{-nd(n)/2} w_{s,r}$ for $s \in S$. Since all $s \in S$ are accepting conversations it follows that the probability that P^{new} wins his game when the verifier has coins r is at least $k_r 2^{-nd(n)}$.

Since the verifier has coins r with probability $2^{-l(n)}$ the probability that P^{new} can win the game is at least $2^{-l(n)-nd(n)/2} \sum_r k_r$. However we know that an optimal prover can only win the game with probability $2^{-2nd(n)}$ and this gives the inequality. □

Consider stage 0. We claim that if Alice has a probability of at least 1/10 of succeeding the there are at least $2^{l(n)-3}$ r's such that $|\beta_{*,r}| \geq 2^{q(n)-l(n)-5}$. If not, one can easily see by Lemma 4.2 that the probability that Alice will succeed at an individual step is at most 1/4 and hence the probability she succeeds in at least 50 of 100 protocols is much smaller than 1/10. Furthermore, since Bob will reject if he ever sees a conversation where the virtual verifier rejected we can assume that there are $2^{q(n)-l(n)-5}$ elements in $\beta_{*,r}$ which lead to an accepting computation. We know by Lemma 5.3 that there cannot be more than $2^{l(n)-4}$ r's for which the subset of $\beta_{*,r}$ which corresponds to conversations where the virtual prover did not cheat can be of size at least $2^{q(n)-l(n)-6}$. Thus for at least $2^{l(n)-4}$ r's there at least $2^{q(n)-l(n)-6}$ elements in $\beta_{*,r}$ for which the virtual prover cheats. Thus the virtual prover cheats in at least $2^{q(n)-10}$ conversations. Since there are only $d(n)/2$ moves by the prover we get:

Lemma 5.4: *There is an i such that a fraction $1/512d(n)$ of the moves y_i by the virtual prover are cheating moves.*

Fix the above i and let us analyze what happens in the protocol during stages 1-5 between Alice and Bob when $j = i$. To do this analysis we will need a classical theorem from Information Theory.

Lemma 5.5: *Let $z^{(j)}, j = 1, 2, \ldots, k$ be independent random variables which takes a discrete set of values. Suppose further that $Pr[z^{(j)} = z_i^{(j)}] \geq \epsilon$ for all possible values $z_i^{(j)}$. Let Z be a random variable which is a direct product of the $z^{(j)}$. Then there is a number H_k such that for any $t \leq \sqrt{k}/5$ there are $\leq 2^{H_k + t|\log\epsilon|\sqrt{k}}$ values Z_i such that:*

1. $2^{-H_k - t|\log\epsilon|\sqrt{k}} \leq Pr[Z = Z_i] \leq 2^{-H_k + t|\log\epsilon|\sqrt{k}}$.
2. $Pr[\exists_i Z = Z_i] \geq 1 - 2e^{\frac{-t^2}{4}}$.

Proof: Let $z_i^{(j)}$ be the possible values of $z^{(j)}$ and let $p_i^{(j)}$ be their probabilities. Let $lz^{(j)}$ be a random variable which takes value $\log p_i^{(j)}$ with probability $p_i^{(j)}$. Let $lZ = \sum_{j=1}^k lz^{(j)}$, $H^{(j)} = \sum_i p_i^{(j)} \log p_i^{(j)}$ and $H_k = \sum_{j=1}^k H^{(j)}$. We prove that lZ is concentrated around H_k.

$$E(e^{\lambda(lZ-H_k)} = \prod_{j=1}^{k} E(e^{\lambda(lz^{(j)}-H^{(j)})}) =$$

$$\prod_{j=1}^{k}(\sum_i p_i^{(j)} e^{\lambda(\log p_i^{(j)} - H^{(j)})})$$

To estimate this quantity we use

Lemma 5.6: *If $\sum_i r_i p_i = \mu$, $\sum_i p_i = 1$ and $\lambda\, max_i |r_i - \mu| \leq \frac{1}{10}$ then*

$$\sum_i p_i e^{\lambda(r_i-\mu)} \leq e^{\lambda^2 \sum_i p_i (r_i - \mu)^2}.$$

Proof: (Lemma 5.6) Assume $\mu = 0$ (otherwise we can set $r'_i = r_i - \mu$). We use $e^{\lambda r_i} \leq 1 + \lambda r_i + (\lambda r_i)^2$ (we know that $|\lambda r_i| \leq 1/10$) and $e^y \geq 1 + y$ valid for any y. This gives

$$\sum_i p_i e^{\lambda r_i} \leq \sum_i p_i (1 + \lambda r_i + (\lambda r_i)^2) =$$

$$1 + \lambda^2 \sum_i r_i^2 p_i \leq e^{\lambda^2 \sum_i r_i^2 p_i}. \quad \square$$

To finish the proof of Lemma 5.5 we use Lemma 5.6 and $\sum_i (\log p_i^{(j)} - H^{(j)})^2 p_i^{(j)} \leq (\log\epsilon)^2$ to get

$$\prod_{j=1}^{k}(\sum_i p_i^{(j)} e^{\lambda(\log p_i^{(j)} - H^{(j)})}) \leq e^{k\lambda^2 (\log\epsilon)^2}.$$

This implies that

$$Pr[|lZ - H_k| \geq t|\,log\,\epsilon|\sqrt{k}] \leq 2e^{k\lambda^2(log\,\epsilon)^2 - \lambda t|\,log\,\epsilon|\sqrt{k}}.$$

Choosing $\lambda = t/2|\,log\,\epsilon|\sqrt{k}$ gives the desired bound and proves Lemma 5.5. □

Let us go back to analyzing the protocol. When B has chosen S_j we can look at the random coins R^V consistent with S_j as a random variable. It satisfies Lemma 5.5 with $k = nd^2(n)q^2(n)$ and $\epsilon = 2^{-q(n)}$ and we will use $t = n$. We get a value for H_k and we will keep that fixed from now on. We will now establish that Alice must fail to convince Bob by a series of facts.

Fact 1: *Alice must choose* $2^{a_j} \geq \frac{|\beta_{S_j}|}{16}$.

Proof: If Alice chooses $2^{a_j} \leq |\beta_{S_j}|/16$ the probability of succeding in an individual round of the first stage of step 3 is by Lemma 4.2 (1), $\leq (3+\sqrt{5})/16 \leq 1/3$. Thus the probability that she will succeed in a majority of the steps is exponentially small. □

Next we derive our first consequence of Lemma 5.5.

Fact 2: *Alice must choose* $b_j \geq H_k - 2q^2(n)d(n)n^{3/2}$.

Proof: Applying Lemma 5.5 with $k = nd^2(n)q^2(n)$, $\epsilon = 2^{-q(n)}$ and $t = n$ we get that there are values R_i^V of R^V such that:

1. $2^{-H_k - q^2(n)d(n)n^{3/2}} \leq Pr[R^V = R_i^V] \leq 2^{-H_k + q^2(n)d(n)n^{3/2}}$.
2. $Pr[\exists_i R^V = R_i^V] \geq 1 - 2e^{-n^2/4}$.

Now suppose Alice chooses $b_j \leq H_k - 2q^2(n)d(n)n^{3/2}$. She must choose $c_j \geq log\,|\beta_{S_j}| - b_j - 19$. Look at the second part of step 3. For Alice to have probability $\geq \frac{1}{4}$ of succeeding, then by Lemma 4.2 there must be 2^{b_j-3} R^V's such that $|\beta_{S_j.R^V}| \geq 2^{c_j-3}$. But this means that the probability of R^V occurring is at least $2^{c_j-3}/|\beta_{S_j}| \geq 2^{-b_j-22} \geq 2^{-H_k+q^2(n)d(n)n^{3/2}}$. But by (2.), the total probability corresponding to such R^V is at most $2e^{-n^2/4}$. Thus there are at most $2^{-n^2/4}|\beta_{S_j}|2^{3-c_j} \leq 2^{b_j+22-n^2/4}$ such R^V and Alice will fail.

Fact 3: *The fraction of* β_{S_j} *corresponding to* R^V*'s that appears with probability* $\leq 2^{-H_k - q^2(n)d(n)n^{3/2}}$ *is* $2e^{-n^2/4}$.

Proof: This is just a restatement of (2.) above.

Next let us look at possible continuations. Let $Y_i = y_i^{(1)}, y_i^{(2)} \ldots y_i^{(nd^2(n)q^2(n))}$ be a potential next move by the virtual prover.

Fact 4: *With probability* $1 - 2^{-cnd(n)q^2(n)}$ *it is true that the fraction of* β_{S_j} *for which the pair* (Y_i, R^V) *does not contain* $nd(n)q^2(n)/1024$ *cheating submoves is* $2^{-cnd(n)q^2(n)}$.

Proof: Take a random complete conversation. It consists of $nd^2(n)q^2(n)$ subconversations each having probability $\geq 1/512d(n)$ of having a cheating move as its j'th move. The expected number of cheating moves is thus $nd(n)q^2(n)/512$. By ordinary estimates for sums of independent variables the probability of having fewer than $nd(n)q^2(n)/1024$ cheating submoves is $2^{-c_1nd(n)q^2(n)}$ for some constant c_1. Thus the expected fraction of β_{S_j} that has $\leq nd(n)q^2(n)/1024$ cheating submoves is $2^{-c_1nd(n)q^2(n)}$. This implies that the probability that a random S_j would give a β_{S_j} for which that fraction of conversations which contain at most $nd(n)q^2(n)/1024$ cheating submoves is $\geq 2^{-c_1nd(n)q^2(n)/2}$ is $2^{-c_1nd(n)q^2(n)/2}$. This proves the fact with $c = c_1/2$. □

Let us consider stage 5 of the protocol after Alice has decided on a d_j. For Alice to have probability 1/4 to succeed there must be 2^{d_j-3} Y_i's each having 2^{b_j-3} R^V's such that each pair (Y_i, R) corresponds to $2^{c_j-d_j-3}/nd^2(n)q^3(n)$ possible runs of M_1.

Using $2^{b_j+c_j} \geq |\beta_{S_j}|2^{-19}$ it is clear that removing a total of $\leq 2^{-n}|\beta_{S_j}|$ occurrences of pairs (Y_i, R) there must still remain 2^{d_j-4} Y_i's each having 2^{b_j-4} R^V's such that each pair (Y_i, R^V) has $2^{c_j-d_j-4}/nd^2(n)q^3(n)$ occurrences. Thus we can remove all elements of β_{S_j} which give an R^V corresponding to a lower probability than prescribed by Fact 3, or which give fewer cheating moves than prescribed by Fact 4.

Now we are ready for the final contradiction. Take any Y_i. We have to count the number of R^V's which satisfy the above two hypotheses. But since all R^V's appear with about equal probability we can replace counting by calculating the probability that a random R^V satisfy these hypotheses. Thus we estimate the probability that Y_i, R^V gives $nd(n)q^2(n)/1024$ cheating pairs. Since the probability that each individual pair y_i, r_i is cheating is 2^{-n} and there are $\binom{nd^2(n)q^3(n)}{\frac{nd(n)q^2(n)}{1024}}$ ways of choosing the places where the cheats should occur, we get the total bound:

$$\binom{nd^2(n)q^2(n)}{\frac{nd(n)q^2(n)}{1024}} 2^{-\frac{n^2d(n)q^2(n)}{1024}} \leq$$

$$(nd^2(n)q^2(n)2^{-n})^{\frac{nd(n)q^2(n)}{1024}} \leq 2^{-\frac{n^2d(n)q^2(n)}{2048}}$$

Since each R^V appears with probability at least $2^{-H_k-q^2(n)n^{3/2}d(n)}$ the number of possible R^V's
is bounded by $\leq 2^{H_k+q^2(n)n^{3/2}d(n)-n^2d(n)q^2(n)/2048}$. Using Fact 2 we see that that this is much fewer than the required 2^{b_j-4} and we have reached a contradiction.

Acknowledgments: We would like to thank Lance Fortnow for valuable comments.

References

[AHG] Aiello, W., J. Hastad, and S. Goldwasser, "On the Power of Interaction," *Proc. of 27th Symposium of the Foundations of Computer Science*, pp 368–379, Toronto, 1986.

[Ba] Babai L., "Trading Group Theory for Randomness," *Proc. of 17th Symposium on the Theory of Computation*, pp 421–429, Providence, Rhode Island, 1985.

[Be] Benor M., private communication, 1986.

[F] Fortnow L., "The Complexity of Perfect Zero-Knowledge," *Proc. of 19th Symposium of the Theory of Computation*, New York, 198?

[GMR] Goldwasser, S., S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proofs," *Proc. of 17th Symposium on the Theory of Computation*, pp 291–305, Providence, Rhode Island, 1985.

[GMW] Goldreich, O., S. Micali, and A. Wigderson, "Prooof that Yield Nothing but their Validity and a Methodology of Cryptographic Protocol Design," *Proc. of 27th Symposium on the Foundations of Computer Science*, pp 174–187, Toronto, 1986.

[GS] Goldwasser, S., and M. Sipser, "Private Coins Versus Public Coins in Interactive Proof Systems," *Proceedings of 18th Symposium on Theory of Computing*, pp 59–68, Berkeley, 1986.

[M] Moran S., private communication, 1986.

[S] Sipser M., "A Complexity Theoretic Approach to Randomness," *Proc. of 15th Symposium on the Theory of Computation*, pp 330–335, Boston, 1983.