Check for updates

The Complexity of Perfect Zero-Knowledge

(extended abstract)

Lance Fortnow*
MIT Math Dept.†
Cambridge, MA 02139

Abstract

A *Perfect Zero-Knowledge* interactive proof system convinces a verifier that a string is in a language without revealing any additional knowledge in an information-theoretic sense. We show that for any language that has a perfect zero-knowledge proof system, its complement has a short interactive protocol. This result implies that there are not any perfect zero-knowledge protocols for NP-complete languages unless the polynomial time hierarchy collapses. This paper demonstrates that knowledge complexity can be used to show that a language is easy to prove.

1 Introduction

Interactive protocols and zero-knowledge, as described by Goldwasser, Micali and Rackoff [GMR], have in recent years proven themselves to be important models of computation in both complexity and cryptography.

Interactive proof systems are a randomized extension to NP which give us a greater understanding of what an infinitely powerful machine can prove to a probabilistic polynomial one. Recent results about interactive protocols have given us an idea of what languages may be efficiently provable in this way.

*The author is supported in part by an Office of Naval Research fellowship, NSF Grant DCR-8602062 and Air Force Grant AFOSR-86-0078.

†Much of this work was done while the author was at the University of California at Berkeley.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specfic permission.

© 1987 ACM 0-89791-221-7/87/0006-0204 75¢

Zero-knowledge interactive protocols give us a good way to determine which languages can be efficiently proven without giving away any details of the proof. This model consists of an infinitely powerful prover trying to convince a polynomial time verifier that a string is in a certain language. Zero-knowledge requires that the verifier not learn any information useful to him as a polytime machine. Goldreich, Micali and Wigderson [GMW] show that if one way functions exist then all languages in NP have zero-knowledge proofs. However, their proof relies on the fact that the verifier has limited power and is unable to invert these one-way functions. A stronger notion is that of perfect zero-knowledge(PZK) which requires that the verifier not learn any additional information no matter how powerful he may be. There are many languages not known to be in BPP or NP∩co-NP, such as graph isomorphism [GMW], which have perfect zero-knowledge proof systems.

Our main theorem says that for any language which has a perfect zero-knowledge protocol, its complement has a single round interactive protocol. Thus PZK⊆co-AM, where AM is the class accepted by one-round Arthur-Merlin games described by Babai [B]. Our result holds in the weaker case where we only require that the verifier which follows protocol will not learn any additional information.

Combining our main theorem with a result of Boppana and Hastad [BH], we get that NP-complete languages do not have perfect zero-knowledge proof systems unless the polynomial time hierarchy collapses to the second level. Thus it is unlikely that the result of [GMW] that NP has zero-knowledge proof systems will extend to perfect zero-knowledge.

Our proof makes use of an approximate upper bound protocol that is of independent interest and that may be useful in completely different contexts. This is in contrast to an approximate lower bound protocol used in [S,B,GS].

The results in this paper do not depend on any unproven cryptographic assumptions.

2 Notation and Definitions

Let P be a probabilistic infinite power Turing machine and V be a probabilistic polynomial time machine that share the same input and can communicate with each other. Let $P \leftrightarrow V$ denote the interaction between P and V. $P \leftrightarrow V(x)$ accepts if after the interaction, V accepts. V's *view of the conversation* between P and V consists of all the messages between P and V and the random coin tosses of V.

P and V form an interactive protocol for a language L if:

1. If $x \in L$ then $\Pr(P \leftrightarrow V(x) \text{ accepts}) > 1 - \frac{1}{p(|x|)}$ for all polynomials p and x large enough.
2. If $x \notin L$ then $\forall P^* \Pr(P^* \leftrightarrow V(x) \text{ accepts}) < \frac{1}{p(|x|)}$ for all polynomials p and x large enough.

Let IP be the class of all languages that are *efficiently provable*, *i.e.*, accepted by an interactive protocol.

A *round* of an interactive protocol is a message from the verifier to the prover followed by a message from the prover to the verifier. AM is the class of languages accepted by bounded round interactive protocols.

The notation for describing protocols follows:

P: These are computations performed by the prover that can not be seen by the verifier. The prover has probabilistic infinite time to make these computations.

$P \rightarrow V$: This is a message from the prover to the verifier.

V: These are computations performed by the verifier that cannot be seen by the prover. These computations must be performed in probabilistic polynomial time.

$V \rightarrow P$: This is a message from the verifier to the prover.

Let M be a simulator for a view of the conversation between P and V. Each run of M will produce:

$$r, \beta_1, \alpha_1, \beta_2, \ldots \beta_k, \alpha_k$$

where the α_i are messages from the prover to the verifier at round i and the β_i are messages from the verifier to the prover, and r is the random coin tosses of V.

Let $P \leftrightarrow V[x]$ denote the distribution of views of conversations between P and V over the random coin tosses of P. $M[x]$ denotes the distribution of views of conversations created by running M on x.

Let $A[x]$ and $B[x]$ be two distributions of strings. $A[x]$ and $B[x]$ are *statistically close* if for any subset of strings S,

$$\left| \sum_{y \in S} \Pr_{A[x]}(y) - \sum_{y \in S} \Pr_{B[x]}(y) \right| < \frac{1}{q(|x|)}$$

for all polynomials q with x large enough. Let p be a probabilistic polynomial time machine that outputs either 0 or 1. $A[x]$ and $B[x]$ are *polytime indistinguishable* if for any p,

$$|\Pr(p(A[x]) = 1) - \Pr(p(B[x]) = 1)| < \frac{1}{r(|x|)}$$

for all polynomials r with x large enough. Note that if $A[x]$ and $B[x]$ are statistically close then they are polytime indistinguishable.

$P \leftrightarrow V$ is *Zero-Knowledge*(ZK) if for any V^* there is a M_{V^*} such that $(\forall x \in L) P \leftrightarrow V^*[x]$ and $M_{V^*}[x]$ are polytime indistinguishable.

$P \leftrightarrow V$ is *Perfect Zero-Knowledge*(PZK) if for any V^* there is a M_{V^*} such that $(\forall x \in L) P \leftrightarrow V^*[x] = M_{V^*}[x]$.

$P \leftrightarrow V$ is *Almost Perfect Zero-Knowledge*(APZK) if for any V^* there is a M_{V^*} such that $(\forall x \in L) P \leftrightarrow V^*[x]$ and $M_{V^*}[x]$ are statistically close.

Interactive Protocols and Zero-Knowledge were introduced in [GMR]. Perfect Zero-Knowledge was described originally in [GMW]. The class AM was introduced by Babai [B] as the class of languages that have one round interactive protocol where V's message consists exactly of his coin tosses. This was shown equivalent to the definition used above by [GS] and [B].

Note that ZK$\supseteq$APZK$\supseteq$PZK. The inclusions are not known to be proper but this paper gives good evidence that ZK$\neq$APZK.

The results in this paper only require a weaker version of zero-knowledge: a simulator only need exist for the given P and V and not when the verifier cheats. For the rest of this paper we will assume we are in this weaker model and $M = M_V$ is the simulator for P and V.

ϵ is used as a small number that will be determined in the final paper.

3 Related Results

Goldwasser and Sipser [GS] have shown that for any language that has an interactive protocol in Q rounds, there is an Arthur-Merlin protocol in $Q + 2$ rounds for that language. Arthur-Merlin protocols are similar to interactive protocols except that the verifier's

messages are just random coin tosses. Babai [B] showed that any bounded round Arthur-Merlin protocol is equivalent to a one round Arthur-Merlin protocol. This is just our class AM. Babai also shows that AM$\subseteq NP^R$ for a random oracle R and also that AM$\subseteq \Pi_2^p$. Mike Sipser pointed out that AM is contained in non-uniform NP.

Boppana and Hastad [BH] show that if co-NP has bounded round interactive proofs then the whole polynomial time hierarchy would be in AM or thus PH$\subseteq \Pi_2^p$ and thus the polynomial time hierarchy collapses to the second level.

Brassard and Crépeau [BC] show perfect zero-knowledge for SAT using a different model for interactive protocols where the prover is a polynomial time machine that knows a satisfying assignment. Our result about perfect zero-knowledge relies on the ability of the prover to have infinite power and thus does not apply to Brassard and Crépeau's model.

4 Showing Sets are Large and Small

In this paper, we will need protocols to show sets are large and small. We do both using Carter-Wegman Universal Hash Functions [CW].

Suppose $\mathcal{X} = \{0,1\}^N$ is a set of strings and $\mathcal{S} \subseteq \mathcal{X}$. Let A be a random binary $N \times \ell$ matrix. Let $f : \mathcal{X} \rightarrow \{0,1\}^\ell$ be the function defined by $f(x) = Ax$. Let $f_\mathcal{S}$ be the restriction of f to $\mathcal{S}$.

If $|\mathcal{S}| \gg 2^\ell$ then $f_\mathcal{S}$ is likely to be onto most of $\{0,1\}^\ell$ and most elements of $\{0,1\}^\ell$ will have many preimages.

If $|\mathcal{S}| \ll 2^\ell$ then the range of $f_\mathcal{S}$ is a small subset of $\{0,1\}^\ell$ and most elements of $f_\mathcal{S}(\mathcal{S})$ have only one inverse in $\mathcal{S}$.

If S is polynomial time checkable then we have the following protocol to show $|\mathcal{S}| \gg 2^\ell$:

V: Pick a random $N \times (\ell + \epsilon)$ matrix A and $y \in \{0,1\}^{\ell+\epsilon}$.

$V \rightarrow P$: A, y

$P \rightarrow V$: $x \in \mathcal{S}$ such that $f(x) = Ax = y$

If $\mathcal{S}$ is not much larger than 2^ℓ then it is unlikely for there to be any x such that $f(x) = y$. However if $\mathcal{S}$ is large then there are likely to be many x so a prover should have no trouble exhibiting such an x that V can verify in polynomial time.

If V has a random element $s \in \mathcal{S}$ that is not known by P then the following protocol works to show $|\mathcal{S}| \ll 2^\ell$:

V: Pick a random $N \times (\ell - \epsilon)$ matrix A.

$V \rightarrow P$: $A, f(s) = As$

$P \rightarrow V$: s

If $\mathcal{S}$ is small then s is likely to be the only element of $\mathcal{S}$ that maps to $f(s)$ thus P can find s. If $\mathcal{S}$ is large then many elements of $\mathcal{S}$ map to $f(s)$,and because s is a random element of $\mathcal{S}$, the prover will have no way of determining which element of $\mathcal{S}$ V has.

Suppose we had two sets $A, B \subseteq \mathcal{X}$ and wanted to show that $|A| \gg |B|$. If A was polynomial time checkable and V has a random element b of B, then this is a protocol to show $|A| \gg |B|$:

P: Pick ℓ such that $|A| \gg 2^\ell \gg |B|$

$P \rightarrow V$: ℓ

$P \rightarrow V$: Prove that $|A| \gg 2^\ell$ as above

$P \rightarrow V$: Prove that $|B| \ll 2^\ell$ as above

Using Carter-Wegman Hashing to show a set is large was introduced by Sipser [S] and used extensively in [S,B,GS]. To the author's knowledge this paper is the first use of an interactive protocol to show a set is small.

5 Main Theorem

We will start with a simple version of the theorem:

Theorem 1 *Let L be a language with a perfect zero-knowledge interactive protocol. Then there exists an interactive protocol accepting $\overline{L}$.*

5.1 Structure of Proof

We are given a prover and verifier (P and V) for the language L and a simulator M that produces views of conversations between P and V and the random coin tosses of V. Note one can simulate the computation of V in polynomial time, checking, for example, whether or not V accepts. On $x \in L$, M produces a view of a conversation from exactly the same probability distribution as when P and V run on x. The key idea of the proof of the theorem is to notice what M might do on $x \notin L$. There is no requirement in the definition of perfect zero-knowledge on what M does on $x \notin L$; however there are three possibilities:

1. M will produce "garbage", something that clearly is not a randomly selected member of $P \leftrightarrow V[x]$.

2. M will produce views of conversations that cause V to reject most of the time.

3. M will produce a simulation that looks valid and causes V to accept. It may not be possible in polynomial time to tell this view from one created by P and V when $x \in L$. However, M must be producing views of conversations from a distribution quite different from the distribution of views between P and V, since in the real views V is likely to reject.

We will create a new prover and verifier, P' and V' that will determine if one of the three cases occur. V' will run M and get a view of a conversation between P and V and r, the random coin tosses of V. V' will check that this view is valid and that V halts accepting. If the view fails this test then it falls in cases 1 or 2 so V' knows that $x \notin L$ and V' accepts. Otherwise V' will send to P' some initial segment of the conversation. P' will then convince V' that the conversation came from a bad distribution by "predicting" r better than P' could have done from a good distribution.

5.2 Example: Graph Isomorphism

Graph isomorphism is a well studied problem that is clearly in NP but not known to be in co-NP. A perfect zero-knowledge proof of graph isomorphism was presented in [GMW]. We will show how the theorem converts this perfect zero-knowledge protocol to an interactive protocol for graph non-isomorphism. This protocol for graph non-isomorphism is identical to the graph non-isomorphism protocol described in [GMW]; our proof, however, shows that the similarity between the two protocols is not coincidental.

Let $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ be a permutation of the vertices of a graph. For a graph $G = (V, E)$ let $(\pi(v_1), \pi(v_2)) \in \pi(E) \Leftrightarrow (v_1, v_2) \in E$. Let $\pi(G) = (V, \pi(E))$.

Two graphs G_1 and G_2 are *isomorphic* if there exists a permutation π such that $\pi(G_1) = G_2$. A perfect zero-knowledge protocol for graph isomorphism suggested by [GMW] works as follows:

P: Generates random permutation π and computes $G = \pi(G_1)$

$P \rightarrow V$: G

$V \rightarrow P$: $i = 1$ or 2 chosen at random

$P \rightarrow V$: π' chosen at random such that $\pi'(G_i) = G$

If $G_1 \cong G_2$ then G will be a permutation of both G_1 and G_2 and P will always be able to find a π'. If $G_1 \ncong G_2$ then G cannot be a permutation of both G_1 and G_2 so at least half of the time V will choose an i such that no π' exists. Thus we have an interactive protocol for graph isomorphism. This protocol also is perfect zero-knowledge.

The simulator M works as follows:

M generates π and i at random and computes $G = \pi(G_i)$ then outputs the following view of a conversation:

r: i

$P \rightarrow V$: G

$V \rightarrow P$: i

$P \rightarrow V$: π

It is easy to verify that when $G_1 \cong G_2$, M produces exactly the same distribution of views of conversations as P and V. Notice what happens when $G_1 \ncong G_2$. The output of M always causes V to accept. Thus when $G_1 \ncong G_2$, M must produce views of conversations from a very different distribution from what P and V produce. In fact whenever $G_1 \ncong G_2$, one can always predict $r = i$ from the G produced by M. This leads to a new interactive protocol between a new prover and verifier, P' and V', for graph non-isomorphism as follows:

V': Generate π and i at random and compute $G = \pi(G_i)$

$V' \rightarrow P'$: G

$P' \rightarrow V'$: i

5.3 The Protocol for $\bar{L}$

We are given a prover and verifier, P and V for a language L and a simulator M that exactly simulates views of conversations between P and V when $x \in L$. A protocol between a new prover and verifier, P' and V', works as follows:

V': Run M and get $r, \beta_1, \alpha_1, \ldots \beta_k, \alpha_k$. V' now checks two things:

1. Check that the conversation is *valid*, *i.e.*, that $r, \alpha_1, \ldots \alpha_k$ will cause V to say $\beta_1, \ldots \beta_k$.
2. Check that the conversation causes V to accept.

If either of these tests fail then V' can be pretty sure that $x \notin L$ so V' quits now and accepts. Otherwise V' continues.
Let i=1.

$V' \rightarrow P'$: β_i, α_i

$P' \rightarrow V'$: STOP and show $|\mathcal{R}_1| \gg |\mathcal{R}_2|$ (as defined below) or
TRY NEXT ROUND (P' must say STOP after some α_j)

$\mathcal{R}_1$ can be thought of as all the possible random strings of V after round j of the protocol. $\mathcal{R}_2$ are the possible random strings of V generated by M. More formally:

Let $\mathcal{R}$ be the set of all possible coin tosses of V.
Let $\mathcal{R}_1 = \{R \in \mathcal{R} | R$ and $\alpha_1, \ldots \alpha_j$ cause V to say $\beta_1, \ldots \beta_j\}$.
Let $\mathcal{R}_2 = \{R \in \mathcal{R} |$ Pr(M outputs $R, \beta_1, \alpha_1, \ldots \beta_j, \alpha_j$ part of a valid, accepting conversation$|M$ produces $\beta_1, \alpha_1, \ldots \beta_j, \alpha_j) \geq \frac{1}{|\mathcal{R}_1|^{1+\epsilon}}\}$.

Note that $\mathcal{R}_2 \subseteq \mathcal{R}_1$ and if $x \in L$ then $\mathcal{R}_2 \approx \mathcal{R}_1$. Also note that $\mathcal{R}_1$ is independent of α_j.

Since $\mathcal{R}_1$ is polynomial time checkable and V' knows a random element of $\mathcal{R}_2$, namely r, we can use the protocol outlined in section 4 to show $|\mathcal{R}_1| \gg |\mathcal{R}_2|$.

5.4 The Protocol Constitutes an Interactive Protocol for $\bar{L}$

To show that this is an interactive protocol for $\overline{L}$, we must show two things:

1. If $(x \in \overline{L})$ then $P' \leftrightarrow V'(x)$ accepts with high probability.
2. If $(x \notin \overline{L})$ then $\forall \hat{P}$ $\hat{P} \leftrightarrow V'(x)$ accepts with very low probability.

We will prove the second statement first since it is the easier of the two to prove.

2. Suppose $x \in L$. Then M will produce views of conversations from exactly the same distribution as P and V. So virtually all conversations produced will be valid and cause V to accept. So after round j (for any j), r will be distributed about uniformly over the possible R's that could have taken V this far. Thus, $\mathcal{R}_2 \approx \mathcal{R}_1$ so no $\hat{P}$ will be able to convince V' that $|\mathcal{R}_1| \gg |\mathcal{R}_2|$.

1. Suppose to the contrary that $x \notin L$ and the protocol does not work. So $\mathcal{R}_2 \approx \mathcal{R}_1$ at all rounds j with high probability. We use this to derive a contradiction by demonstrating that $P \leftrightarrow V$ is not an interactive proof system for L by presenting a prover P^* that will convince V (the original verifier) that $x \in L$.

At round j suppose the conversation so far has been $\beta'_1, \alpha'_1, \ldots \beta'_j$. P^* works as follows:

P^*: Run M which outputs $r, \beta_1, \alpha_1, \ldots \beta_k, \alpha_k$. Check that this is a valid accepting conversation. If not, try again. See if $\beta_1, \alpha_1 \ldots \beta_j = \beta'_1, \alpha'_1, \ldots \beta'_j$. If not, try again.

$P^* \rightarrow V$: α_j

At round j when P^* has a conversation from M that matches the conversation so far, $\mathcal{R}_1$ is the set of possible random coin tosses of V. When P^* says α_j then $\mathcal{R}_2$ is the set of coin tosses of V that will still keep V heading towards an accepting path. Since $\mathcal{R}_2 \approx \mathcal{R}_1$, this will happen with high probability. Even after a polynomial number of rounds, V still has a decent chance of accepting x. This cannot be since $x \notin L$ and $P \leftrightarrow V$ is an interactive protocol for L.

Note that P^* may require exponential expected time to complete its part of the protocol but in our model an infinitely powerful P^* is allowed.

6 Extensions and Corollaries

Theorem 2 *Suppose $P \leftrightarrow V$ is an interactive protocol for a language L and there is a probabilistic polynomial time simulator M such that $M[x]$ is statistically close to $P \leftrightarrow V[x]$. Then there is a single round interactive protocol for the complement of L.*

Proof This extends the main theorem in two ways. First, we do not require $M[x] = P \leftrightarrow V[x]$ just that they be statistically close. One can check the proof in the previous section and notice that statistically close is good enough.

Second, we would like to get a single round protocol for the complement of L. Notice that in the protocol given above the number of rounds is dependent on when P' decides to say STOP. To get bounded rounds we must make the following change to the protocol:

V': Run M k^3 times independently and get k^3 views of conversations and check that each conversation is valid and accepting.

$V' \rightarrow P'$: For $1 \leq i \leq k^3$ send the first $i \bmod k$ rounds of the ith conversation.

$P' \rightarrow V'$: Pick any conversation j and show $|\mathcal{R}_1| \gg |\mathcal{R}_2|$ for the view of that conversation.

It is not hard to verify that the above proof still works for this new protocol. Once we have bounded rounds we apply the theorems of [B,GS] which imply that all bounded round protocols can be made into single round protocols.

Some trivial corollaries that follow from results that are described in section 3:

Corollary 1 *If $L \in APZK$ then $\overline{L} \in AM$.*

Corollary 2 *If $L \in APZK$ then $L \in co\text{-}NP^R \subseteq$ non-uniform co-NP where R is a random oracle.*

Corollary 3 *If any NP-complete language has an almost perfect zero-knowledge interactive protocol then the polynomial time hierarchy collapses to the second level.*

Corollary 4 *If L and $\overline{L}$ both have almost perfect zero-knowledge interactive protocols with possibly unbounded rounds then $L \in (NP \cap co\text{-}NP)^R$ where R is a random oracle.*

Corollary 5 *If there are one-way functions and the polynomial time hierarchy does not collapse then $NP \subseteq ZK$ but $NP \not\subseteq APZK$ so $ZK \neq APZK$.*

7 Open Problems

There are several interesting problems remaining, including:

- What is the relationship between PZK and APZK?
- Are complement of perfect or almost perfect zero-knowledge languages themselves perfect zero-knowledge in any sense?
- Can the same techniques be used to show that any perfect zero-knowledge protocol has a bounded round interactive protocol?

8 Acknowledgements

The author would like to express his gratitude to his advisor, Mike Sipser, for his support and encouragement.

The author would also like to thank Mike, Silvio Micali, Oded Goldreich, Joan Feigenbaum, Paul Beame and Eric Schwabe for their useful comments on this paper.

Also, the author would like to thank New York City—his birthplace, hometown and site of this, his first conference presentation.

References

[B] Babai, L., "Trading Group Theory for Randomness", *Proc. 17th STOC*, 1985, pp. 421-429.

[BH] Boppana, R. and J. Hastad, "Does co-NP Have Short Interactive Proofs?", *IPL*, to appear.

[BC] Brassard, G. and C. Crépeau, "Non-Transitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond", *Proc. 27th FOCS*, 1986, pp. 188-195.

[CW] Carter, J.L. and M.N. Wegman, "Universal Classes of Hash Functions", *JCSS* **18** 2, 1979, pp.143-154.

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

[GMR] Goldwasser, S., S. Micali and C. Rackoff, "The Knowledge Complexity of Interactive Proof-Systems", *Proc. 17th STOC*, 1985, pp. 291-304.

[GS] Goldwasser, S. and M. Sipser, "Private Coins versus Public Coins in Interactive Proof Systems", *Proc. 18th STOC*, 1986, pp. 59-68.

[S] Sipser, M., "A Complexity Theoretic Approach to Randomness", *Proc. 15th STOC*, 1983, pp. 330-335.