Zero-Knowledge With Finite State Verifiers (Extended Abstract)

Cynthia Dwork Larry Stockmeyer

IBM Almaden Research Center
San Jose, CA 95120

Abstract. We initiate an investigation of interactive proof systems (IPS's) and zero knowledge interactive proof systems where the verifier is a 2-way probabilistic finite state automaton (2pfa). Among other results, we show:

1. There is a class of 2pfa verifiers and a language L such that L has a zero knowledge IPS with respect to this class of verifiers, and L cannot be recognized by any verifier in the class on its own;

2. There is a language L such that L has an IPS with 2pfa verifiers but L has no zero knowledge IPS with 2pfa verifiers.

1. Introduction

Issues in complexity theory and cryptography motivated Babai [1] and Goldwasser, Micali, and Rackoff [7] to introduce the concept of an interactive proof system. Speaking informally, an *Interactive Proof System* (IPS) for membership in a language L is a two-party protocol whereby a "prover" convinces a "verifier" that elements $x \in L$ are actually in L. The concept is interesting only if the verifier is not itself sufficiently powerful to recognize L.

To date, almost all research in interactive proof systems has dealt with the case that the verifier is a probabilistic Turing machine (ptm) which runs in polynomial time. Due to the present lack of understanding of the power of polynomial time computation, many previous results depend on unproven assumptions, typically that a certain problem is not in P or that a certain cryptosystem cannot be broken in polynomial time. If the given assumptions are false, then either the proof becomes invalid or the result becomes trivial. For example, the important and powerful result that any language in NP has a zero knowledge IPS [6] would become unproven if secure probabilistic encryption schemes do not exist, and would become vacuous if P = NP.

The ability to prove lower bounds is crucial to understanding the structure of the class of languages with interactive proof systems. We therefore restrict the class of verifiers, namely, to 2-way probabilistic finite state automata (2pfa). We have obtained a number of results on 2pfa's and IP(2pfa), the class of languages with interactive proof systems in which the verifier is a 2pfa, examining public coins, private coins, and zero knowledge proof systems. ([4] contains a preliminary report of these results, including all proofs.) For the remainder of this abstract we restrict our attention to zero knowledge interactive proofs, noting only that the class IP(2pfa) is quite rich, despite the restricted power of the verifier, containing, for example, any language recognizable by a deterministic exponential time Turing machine.

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 71-75, 1990.

2. Definitions

Our definition of an interactive proof parallels the one used in previous papers on interactive proofs where the verifier is a polynomial-time bounded ptm, for example, [7,6], and the one given by [3] in a more general setting. The main difference in our case is that the verifier is a 2-way probabilistic finite state automaton (2pfa). A 2pfa consists of a probabilistic finite state control and a 2-way head which reads the input string. Transition probabilities are assumed to be rational. In addition, the verifier can communicate with a prover which sees the same input. In our case, the communication is done via a single communication cell which can hold a single symbol from some finite communication alphabet. The prover writes a symbol in the cell only in response to a symbol written by the verifier. At some point in the interactive computation, the verifier can halt and either accept or reject. The prover-verifier pair (P, V) is an *interactive proof system for the language L with error probability ε* if

1. for all $x \in L$, $(P, V)(x)$ accepts with probability at least $1 - \varepsilon$, and
2. for all $x \notin L$ and all provers P^*, $(P^*, V)(x)$ rejects with probability at least $1 - \varepsilon$.

Let IP(2pfa) be the class of languages L such that L has an interactive proof system with error probability $\varepsilon < 1/2$.

Let 2PFA denote the class of languages recognized by 2pfa's with error probability $\varepsilon < 1/2$. Equivalently, 2PFA is IP(2pfa) restricted to IPS's (P, V) where P and V do not communicate (so the prover can be empty).

In some results we will want to talk about the expected or worst-case *time complexity* of an IPS (P, V), defined to be the expected (averaged over all random choices made by V and P) or worst-case number of steps taken by the verifier before halting and measured as a function $T(n)$ of the length n of the input.

A *sweeping 2pfa* is a 2pfa restricted so that the input head can switch direction only when reading an endmarker. In any computation, the input head alternately sweeps across the input from left to right, then from right to left, and so on.

We shall also use a more general form of recognition called separation. Let M be an IPS or a 2pfa, and let A and B be sets of words with $A \cap B = \emptyset$. Then M *separates A and B* if there is some constant $\varepsilon < 1/2$ (the error probability) such that, for all $x \in A$, $M(x)$ accepts with probability at least $1 - \varepsilon$, and for all $x \in B$, $M(x)$ rejects with probability at least $1 - \varepsilon$ (we do not care about the behavior of M on inputs not in A or B).

We temporarily defer the definition of "zero knowledge" interactive proof system.

2.1. An Example

If x is a string, let x^R be x written backwards. Define

$$Palindromes = \{ x \in \{0,1\}^* \mid x = x^R \}.$$

We describe an IPS (P, V) for *Palindromes* with error probability ε for any constant $\varepsilon > 0$. If x is a palindrome, the interaction involves k iterations, where $k = \lceil \log_2(1/\varepsilon) \rceil$. On each iteration, the prover P sends x to the verifier one symbol at a time. At the start of each iteration, the verifer V (privately) tosses a fair coin. Letting w denote the string received from the prover during this iteration, if the outcome of the coin toss is "heads" then V checks that $w = x$ and rejects if not. If the outcome is "tails" then V checks that $w = x^R$ and rejects if not. If the check succeeds for all k iterations, then V accepts. It is easy to see that (P, V) is an IPS for *Palindromes* with error probability ε. This shows:

Theorem 2.1. *Palindromes* $\in$ *IP(2pfa). Moreover, for any error probability* $\varepsilon > 0$, *there is an IPS for Palindromes where the verifier is a sweeping 2pfa which runs in worst-case time* $O(n)$.

This theorem contrasts with the following impossibility result.

Theorem 2.2. *Palindromes* $\notin$ *2PFA.*

In fact, we prove a somewhat stronger result, from which the theorem follows. Theorem 2.2 is particularly interesting in light of Freivalds' result [5] that 2PFA contains certain nonregular sets, such as $\{ 0^n 1^n \mid n \geq 1 \}$.

3. Zero Knowledge Interactive Proof Systems

3.1. Old and New Definitions

Informally, an interactive proof system (P, V) for a language L is *zero knowledge* if for any input $x \in L$ and any verifier V^*, the only information which V^* can get from P during their interaction is the single bit of information that x belongs to L. Previous papers, e.g. [7], considered zero-knowledge only for ptime-ptm verifiers; we generalize the definition to an arbitrary class of verifiers as follows. Fix some class $\mathcal{V}$ of verifier machines, for example, 2pfa's or polynomial-time ptm's. Let $(\emptyset, \mathcal{V})$ be the subclass of machines in $\mathcal{V}$ that do not communicate with the prover (the symbol $\emptyset$ in this notation should be a reminder that the prover is empty). The interactive computation of $(P, V^*)(x)$ defines a distribution of conversations between P and V^*. The IPS (P, V) is *zero knowledge* if for any verifier $V^* \in \mathcal{V}$ there is an $M_{V^*} \in (\emptyset, \mathcal{V})$ such that, for all $x \in L$, $M_{V^*}(x)$ produces a distribution of conversations which is "close" to the distribution produced by $(P, V^*)(x)$.

At first glance, it would appear that the IPS (P, V) for palindromes described above is perfect zero knowledge according to this definition. On input x, the conversation consists of the prover sending x to the verifier several times, and obviously a 2pfa can produce this conversation alone. On an intuitive level, however, this IPS is clearly not zero knowledge for the following reason. Let A be the set of "double palindromes", i.e., the set of palindromes of the form ww^R where w is itself a palindrome, and let B be the set of palindromes not in A. It is not hard to see that there is a 2pfa V^* such that (P, V^*) separates A and B. On input x, V^* first checks that $|x|$ is even and rejects if not. Then starting from the left endmarker, V^* moves its head two to the right for every symbol sent to it by the prover until the right endmarker is reached. At this point, P has finished sending w and is ready to send w^R to V^*, where $x = ww^R$. So V^* is now in a position to compare w with w^R. Since we can show that no 2pfa separates A and B, it is clear that P is giving V^* some extra information which it cannot get by itself.

This suggests the following definition of zero knowledge which we call "recognition zero knowledge" to distiguish it from previous definitions.

Let $\mathcal{V}$ be a class of verifier machines. Let (P, V) be an IPS for the language L where $V \in \mathcal{V}$. Then (P, V) is a *recognition zero knowledge IPS for* L *with* $\mathcal{V}$ *verifiers* if, for any $V^* \in \mathcal{V}$ and any $A, B \subseteq L$ with $A \cap B = \emptyset$ such that (P, V^*) separates A and B, there is an $M_{V^*} \in (\emptyset, \mathcal{V})$ such that M_{V^*} separates A and B.

This is a fairly weak definition, in the sense that if a language has no recognition zero knowledge IPS then it has no zero knowledge IPS in a strong intuitive sense.

3.2. Languages Having No Zero Knowledge IPS

We first consider the palindrome language *Palindromes* defined in §2.1. We are able to show that the ability of a V^* to get extra information from the prover is not a property just of the particular IPS (P, V) described in §2.1. It is an inherent property of *Palindromes*.

Theorem 3.1. *There is no recognition zero knowledge IPS for Palindromes with 2pfa verifiers. This remains true with 2pfa verifiers which run in either polynomial worst-case time or polynomial expected time.*

By a similar proof, we can show that the graph isomorphism problem has no recognition zero knowledge IPS with 2pfa verifiers. This result contrasts with the situation for polynomial-time ptm verifiers, where graph isomorphism does have a (recognition) zero knowledge IPS [6]. We remark that the graph isomorphism problem does have an IPS with a 2pfa verifier.

3.3. A Language With a Recognition Zero Knowledge IPS

That the graph isomorphism problem has no (recognition) zero knowledge IPS with 2pfa verifiers suggests that techniques which have been used to obtain zero knowledge IPS's with ptime-ptm verifiers will not extend to 2pfa verifiers. In fact, we have no example of a language $L \notin$ 2PFA which has a recognition zero knowledge IPS with 2pfa verifiers. With 2pfa verifiers restricted to a certain class $\mathcal{R}$, however, we do have such an example. Let $\mathcal{R}$ denote the class of sweeping 2pfa's that halt in polynomial expected time.

Theorem 3.1, showing that there is no recognition zero knowledge IPS for palindromes, also holds with $\mathcal{R}$ verifiers. It is interesting to contrast this latter result with the result obtained next, that the unary version of palindromes has a recognition zero knowledge IPS with $\mathcal{R}$ verifiers. The unary version of palindromes is the language

$$Upal = \{\, 0^n 1^n \mid n \geq 1 \,\}.$$

Greenberg and Weiss [8] show that *Upal* cannot be recognized by any 2pfa which runs in polynomial expected time; in particular, *Upal* is recognized by no machine in $\mathcal{R}$, so the next result is not vacuous.

Theorem 3.2. *There is a recognition zero knowledge IPS for Upal with $\mathcal{R}$ verifiers.*

Actually, we prove a stronger result from which Theorem 3.2 follows immediately. We describe an IPS (P, V) for *Upal* with the following property. For any V^* and any $\varepsilon < 1/2$, let A (B) be the set of integers n such that $(P, V^*)(0^n 1^n)$ accepts (rejects) with probability at least $1 - \varepsilon$. Then there is a set C of integers such that C separates A and B (i.e., $A \subseteq C$ and $B \cap C = \emptyset$) and $\{\, 1^n \mid n \in C \,\}$ is regular. Our proof of this fact differs from previous proofs of zero knowledge in a significant way. Whereas previous proofs involved a simulation which used V^* as a "black box", our proof uses the internal structure of V^* in an essential way. This proof draws upon several facts from the theory of Markov chains.

4. Related Work

Other results on interactive proof systems with restricted verifiers appear in [2] and [3]. In these papers Condon and Ladner considered the case in which the verifier is restricted to

run in space logarithmic in the length of the input, but they did not address the question of zero knowledge.

More recently, Kilian [9], adopting a defintion of zero knowledge based on the one presented here, has shown that, for verifiers which use logarithmic space and polynomial time, every language which has an IPS also has a zero knowledge IPS; no unproved assumptions are needed to obtain this result.

Note

The authors thank the program committee of CRYPTO '88 for inviting this paper to the conference.

References

[1] L. Babai, Trading group theory for randomness, *Proc. 17th ACM Symp. on Theory of Computing* (1985), pp. 421–429.

[2] A. Condon, Computational models of games, Ph.D. Thesis, Tech. Report 87-04-04, Computer Science Dept., University of Washington, Seattle, WA, 1987.

[3] A. Condon and R. Ladner, Probabilistic game automata, *Proc. Conference on Structure in Complexity Theory, Lecture Notes in Computer Science*, Vol. 223, Springer-Verlag, New York, 1986, pp. 144–162.

[4] C. Dwork and L. Stockmeyer, Interactive proof systems with finite state verifiers (preliminary report), IBM Research Report RJ 6262 (1988).

[5] R. Freivalds, Probabilistic two-way machines, *Proc. International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science*, Vol. 118, Springer-Verlag, New York, 1981, pp. 33–45.

[6] O. Goldreich, S. Micali, and A. Wigderson, Proofs that yield nothing but their validity and a methodology of cryptographic protocol design, *Proc. 27th IEEE Symp. on Foundations of Computer Science* (1986), pp. 174–187.

[7] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *Proc. 17th ACM Symp. on Theory of Computing* (1985), pp. 291–304.

[8] A. G. Greenberg and A. Weiss, A lower bound for probabilistic algorithms for finite state machines, *J. Comput. Syst. Sci.* 33 (1986), pp. 88–105.

[9] J. Kilian, Zero-knowledge with log-space verifiers, *Proc. 29th IEEE Symp. on Foundations of Computer Science* (1988), to appear.