Efficient Identification Schemes Using Two Prover Interactive Proofs

Michael Ben-Or
Hebrew University

Shafi Goldwasser*
MIT

Joe Kilian†
MIT

Avi Wigderson
Hebrew Universoity

Abstract

We present two efficient identification schemes based on the difficulty of solving the subset sum problem and the circuit satisfiability problem. Both schemes use the two prover model introduced by [BGKW], where the verifier (e.g the Bank) interacts with two untrusted provers (e.g two bank identification cards) who have jointly agreed on a strategy to convince the verifier of their identity. To believe the validity of their identity proving procedure, the verifier must make sure that the two provers can not communicate with each other during the course of the proof process. In addition to the simplicity and efficiency of the schemes, the resulting two prover interactive proofs can be shown to be perfect zero knowledge, making no intractability assumptions.

1 Introduction

Ben Or, Goldwasser, Kilian and Wigderson [BGKW] introduced the idea of multi-prover interactive proofs, in order to show how to achieve perfect zero-knowledge interactive proofs for all of IP without using any intractability assumptions.

A multi-prover interactive proof is an extension of an interactive proof. Instead of one prover attempting to convince the verifier that x (the input string) is in a language, the prover consists of two separate agents (or rather two provers) who jointly agree on a strategy to convince the verifier that x is in the language. Although the provers can agree on a strategy before they talk to the verifier, once the interaction

* Supported in part by NSF grant CCR-86-57527, DARPA Contract N00014-89-J-1988, and by a US Israel binational grant

† Supported in part by NSF postdoctoral fellowship and NSF grant CCR-86-57527

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 498-506, 1990.

with the verifier starts they can no longer send each other messages or see the messages exchanged between the verifier and the "other prover".

The main novelty of this model is that the verifier can check interactions with the provers against each other. This allowed [BGKW] to prove that anything provable in this model has a *statistical* zero-knowledge proof, without any intractability assumptions.[1] Of more practical significance, they give a direct, efficient proof that membership in any NP language can be done in *perfect* zero-knowledge. The zero-knowledge protocol for NP proposed in [BGKW] is an adaptation to the two-prover model of any of the known zero-knowledge proofs for NP-complete problems in the one-prover model.

Essentially, the cryptographic encryption schemes used by the protocols of [GMW], [Bl], and [Sh], are replaced by new commit and reveal protocols. In the commit protocol, the first (designated) prover commits to a bit value such that the verifier can not (information theoretically) distinguish between a commitment to a 0 and commitment to a 1. In a reveal protocol, the second prover reveals to the verifier the value committed to by the first prover. The probability that the provers can cheat and reveal a different bit to the verifier than the one committed to, can be made negligible. The commit and reveal protocols consist of a few addition operations which are of trivial complexity. No other operations such as large modular multiplications are necessary.

We note that the overall efficiency of our protocols can be increased by using efficient weaker commital protocol in which the the provers have a non-negligible probability of error.

The application of zero-knowledge interactive proofs to identification schemes has been demonstrated in [FS], [MS], [GQ] among others.

In this paper, we suggest applications of the two prover model to the area of identification schemes. We propose two identification schemes which are much more efficient than those known for the one prover model. The first is based on the intractability of the subset sum problem, and implements a variant of a protocol due to Shamir [Sh]. The second is based on the intractability of the circuit satisfiability problem, and implements a variant of protocols due to Brassard–Chaum–Crépeau [BCC] and Impagliazzo-Yung [IY]. The verifier, which in this case of the identification application is the central computer (or a local center such as an ATM machine), receives two cards (or one card with two physically separated CPU's) and interacts with them to receive a proof that the cards belong to a legal and valid user. It is up to the verifier to ensure the cards can not communicate with each other while interacting with him.

2 Definitions and Background

In this section we review the definitions of multi-prover interactive proofs and the theorem of [BGKW] that all NP languages have perfect zero-knowledge multi-prover

[1] Authors' note: More recently, this proof has been strengthened to achieve perfect zero-knowledge (to appear in the journal version of [BGKW].)

interactive proofs.

Let P_1, P_2 be computationally unbounded Turing machines and V be a probabilistic polynomial time Turing machine. Machines P_1, P_2, and V each have a read-only input tape, a work tape, communication tapes on which V and P_i can write messages for each other, and a random tape. In addition, P_1 and P_2 share the same random tape.

Definition: Let P_1, P_2, V have access to the same input tape. We call (P_1, P_2, V) a *two prover interactive protocol.*

Definition: Let $L \subset \{0,1\}^*$. We say that L has a *two prover interactive proof system* if there exists a probabilistic polynomial time Turing machine V such that:

1. There exists P_1, P_2 such that (P_1, P_2, V) is a two-prover interactive protocol and for all $x \in L$, $prob(V \text{ accepts } x) = 1$.
2. For all P_1, P_2 such that (P_1, P_2, V) is a two-prover interactive protocol, for all $x \notin L$,
$$prob(V \text{ accepts } x) \leq \frac{1}{|x|^t},$$
for all constant t, and for all sufficiently large x.

If the above conditions hold, we call (P_1, P_2, V) from Condition 1 a *multi-prover interactive proof* for L.

Definition: Let (P_1, P_2, V) be an interactive protocol. Let $View_{P_1,P_2,V}(x)$ denote the verifier's view during the protocol, i.e the sequence of messages exchanged between the verifier and the two provers and the verifier's coin tosses. We say that a two-prover interactive protocol (P_1, P_2, V) is *perfect zero knowledge* for a language L if for all V^* there exists a probabilistic Turing machine M such that for all $x \in L$, $M(x)$ is identically distributed to $View_{P_1,P_2,V^*}(x)$ and $M(x)$ terminates in expected polynomial time. We say that L has a *perfect zero knowledge interactive proof* if there exists an interactive protocol (P_1, P_2, V) which is an interactive proof for L and perfect zero-knowledge for L.

The following theorem is instrumental to prove the correctness and security of our authentication schemes.

Theorem [BGKW] : For every $L \in NP$, L has a perfect zero knowledge two prover interactive proof.

3 Subset Sum Based Authentication Scheme

In [Sh] a zero-knowledge proof for 0/1 modular knapsack was presented for the one-prover model. We use of the ideas in [Sh] for our first two-prover identification scheme.

Throughout this section we refer to the verifier as V, and to the provers as P_1, P_2.

For concreteness sake we suggest the reader envisage the verifier V as an ATM machine, and the two provers P_1, P_2 as a pair of cards issued to every customer. The cards and the ATM have a common input, which the Bank gave the customer. When the customer logs in, this information is visible to the Bank. Ordinarily, the bank stores pairs (user-name, common input.)

We let P_1 and P_2 share a random string $R = r_1 r_2 ... r_k$ where $r_i \in \{0, 1, 2\}$ chosen at random and k is a polynomial in the number of identifications R will ever be used for. R can be thought of as either truly random (as we do for the sake of this abstract) in which case P_1, P_2 change it on their own after k authentications, or an outcome of a pseudo random number generator.

We first need to introduce two protocols called *commit* and *reveal.*

Let σ_0 be the identity function and $\sigma_1(0) = 0, \sigma_1(1) = 2, \sigma_1(2) = 1$.

Commit (b,j): (b denotes the bit being committed to, and j how many commits were performed thus far.)

1. V flips a coin $c \in \{0, 1\}$ and send c to P_1
2. P_1 computes $v_j = \sigma_c(r_j) + b \bmod 3$ and sends v_j to V.
3. V stores (j, c, v_j).

Reveal (j): (reveal the j-th bit which was committed to)

1. P_2 sends r_j to V
2. V looks up its stored values, (j, c, v_j), and computes

$$b = v_j - \sigma_c(r_j) \bmod 3.$$

Claim 1: Let $c, g \in \{0, 1\}$. If $prob(r = i) = \frac{1}{3}$ for $i = 0, 1, 2$, then

$$prob(b = g | v = \sigma_c(r) + b \bmod 3) = prob(b = g) = \frac{1}{2}.$$

proof: see [BGKW].

Claim 2: Let $prob(c = 0) = prob(c = 1) = \frac{1}{2}$. Then $\forall \epsilon \geq 0$: if for $b \in \{0, 1\}$,

$$prob(P_2 \text{ successfully reveals } b) \geq 1 - \epsilon,$$

then,

$$prob(P_2 \text{ successfully reveals } \bar{b}) \leq \frac{1}{2} + \epsilon.$$

proof: see [BGKW].

Note then that the probability with which the prover can cheat is bounded by the,

$$\max\{\min\{1 - \epsilon, \frac{1}{2} + \epsilon\}\} \leq \frac{3}{4},$$

which is achieved at $\epsilon = \frac{1}{4}$.

We are now ready for the subset sum authentication protocol.

The input to (P_1, P_2, V) protocol is an instance of the subset sum problem denoted by the tuple $(w_i, 1 \leq i \leq n, T, t)$ where the weights w_i are picked from a range $[1, S_n]$, where n is the security parameter of the application and t denotes the number of w_i in the subset in question.

Prover P_1 has as a private input an index set $J \subset \{1, ..., n\}$ such that $\sum_{i \in J} w_i = T$ and $|J| = t$.

The Identification Protocol

The following protocol for the above variant of subset sum is similar to [Sh]. Let language L be the set of tuples of integers,

$$(w_1, \ldots, w_n, T, t),$$

such that there exists a set $J \subseteq [1, n]$ such that,

- $|J| = t$, and,
- $\sum\limits_{i \in J} w_i = T$.

In the protocol below, let S_n denote a strict upper bound on $\sum w_i$.

1. P_1 permutes the w_i at random and accordingly the set J. Denote the permuted values as w_i', J' such that $\sum_{i \in J'} w_i' = T$. P_1 and V now run a commit protocol on secrets A, B, C, D, E (defined below) using the commit protocol (defined above).

    - $A = \{r_i | 1 \leq i \leq n\}$ where r_i's are picked at random from $[1, S_n]$.
    - $B = \{w_i' | 1 \leq i \leq n\}$.
    - $C = \{s_i = w_i' + r_i \bmod S_n | 1 \leq i \leq n\}$.
    - $D = J'$.
    - $E = \sum_{j \in J'} r_i \bmod S_n$.

2. V uniformly chooses $t \in \{1, 2, 3\}$ and sends t to P_2.

3. Using the reveal secrets protocol defined above between P_2 and V:

    - if $t = 1$, P_2 reveals to V committed secrets A, B and C.
    - if $t = 2$, P_2 reveals to V committed secrets C, D and E.
    - if $t = 3$, P_2 reveals to V committed secrets A, D and E.

4. V checks that indeed

- if $t = 1$, $s_i = r_i + w'_i \bmod S_n$ for all i
- if $t = 2$, $\sum_{i \in D} s_i = E + T \bmod S_n$
- if $t = 3$, $\sum_{i \in D} r_i = E \bmod S_n$.

Now to get a high chance of correctness, this protocol is repeated k times where k is the security parameter, chosen by the parties who are running the protocol.

Theorem 1: Let C be an illegal user pretending to be the legal provers P_1, P_2. Under the assumption that solving the subset-sum problem is hard, for all k, for all $t > 0$, for all sufficiently large n,

$$prob(C \text{ cheats successfully}) \leq (\frac{11}{12})^k + \frac{1}{n^t}.$$

(the $\frac{1}{n^t}$ error accounts for the probability that C solves the subset sum instance.)

Theorem 2: The above protocol is a perfect-zero-knowledge multi-prover interactive proof for the subset sum language.

3.1 Efficiency

The new scheme is more efficient than any other identification scheme thus far proposed based on the theory of zero-knowledge. All other schemes can be classified into two classes: they are either based on the factoring intractability assumption(such as Fiat-Shamir) which we call Type 1, or they are based on zero-knowledge proofs for NP-complete problems which we call Type 2. The identification schemes of Type 1 all use as a basic operation large modular multiplications. The identification schemes of Type 2 all use evaluations of one-way functions, for which the only known suggested implementations use as primitive operations large modular multiplications. Thus, a bottle neck for efficiency is the ability to perform fast modular multiplications.

In our scheme, no such operations are necessary. The primitive operations in the commit and reveal protocols are simple additions modulo 3, which is of trivial complexity.

In terms of number of rounds, [BGKW] have already shown that the scheme remains perfect zero-knowledge even if many executions of the protocol are performed in parallel. An interesting open question is that of analyzing the confidence amplification afforded by running protocols in parallel.

3.2 How to Choose Subset Sum Instances

We choose the subset sum problem as suitable for our purpose, since it is easy to generate instances of it at random, and the complexity of checking that indeed a specific subset of the weights adds up to a target, given a description of the subset, is quite inexpensive.

However, one must be careful in choosing instances of the subset sum problem so that the instances are hard ones to solve. In the above protocol we did not address this issue. This of course is not sufficient. The verifier (Bank) should choose the subset-sum instance among the believed to be hard high density instances, as widely studied in the literature by Lagarias and Odlyzko [LO] and others.

We proceed to suggest one more implementation of an authentication scheme based on the multi-prover model, whose security is based on the circuit satisfiability problem.

4 A Circuit Satisfiability Authentication Scheme

As mentioned before, subset sum is an NP complete problem. In principle, any identification system based on possessing a witness to an arbitrary NP set may be reduced to this identification scheme. However, such reductions may be extremely inefficient, wiping out the efficiency gains obtained by using our system. This motivates the consideration of other NP complete problems for use in identification schemes.

In this section, we briefly describe a second protocol for identification, based on the circuit satisfiability problem. The circuit satisfiability problem is as follows: Given a boolean circuit C, with a single output, is there a setting of the input bits such that C will output a 1? This problem is eminently suitable for reductions from other NP languages, since the procedure for checking a witness is fairly easy to write down as a circuit.

The work of [BCC] and [IY] gives a protocol for the circuit satisfiability problem with the following properties.

1. Given a circuit C of size $s(n)$, each iteration of the protocol requires the prover to commit $\Theta(s(n))$ bits to the verifier. For cryptographic based implementations of this scheme, the prover must send $\Theta(s(n)k)$ bits to the verifier, where k is the security parameter being used.

2. If C is satisfiable, and the prover obeys the protocol, then the verifier will always accept. If C is not satisfiable, then the verifier will reject with probability at least $1/2$, for each iteration of the protocol.

A more recent protocol, given in [KMO], allows for an asymptotically more communication efficient zero-knowledge protocol for circuit satisfiability. Instead of sending $\Theta(s(n)k)$ bits per iteration of the protocol, the prover need only send $\Theta(s(n) + k^2)$ bits.

Using a two card system, we can in fact create a protocol which only requires a total of $\Theta(s(n))$ bits of communication per iteration. Furthermore, iterations can be executed in parallel.

We simply use the circuit construction of [BCC] or [IY] modified to use our commit and reveal schemes of section 4 whenever a commitment is called for.

We also note that as a by product of the type of simulation that is done in the two-prover model (no rewinding of the simulator tape is necessary), many interactions of the modified [BCC] and [IY] protocols can be run in parallel and remain perfect zero-knowledge.

The protocol we obtain has the the following specifications:

1. Given a circuit C of size $s(n)$, each iteration of our protocol requires the provers to commit $\Theta(s(n))$ bits to the verifier. This is accomplished by sending a total of only $\Theta(s(n))$ bits to the verifier.

2. If C is satisfiable, and the provers obey the protocol, then the verifier will always accept. If C is not satisfiable, then the verifier will reject with probability at least 1/8, for each iteration of the protocol. (by Section 4, Claim 2)

We note that the rejection probability for an iteration of a bad proof is now only 1/8, instead of the factor of 1/2 of the original scheme. This is because the verifier will reject only if the provers are forced to decommit a bit which would reveal their cheating if correctly decommitted (which occurs with probability $\frac{1}{2}$), and in this case the verifier detects them trying to decommit a value other than what they originally committed with probability at most $\frac{1}{4}$. Recall, whereas in the ideal abstraction for bit-commital, the provers cannot change the value of a decommitted bit, in our system they can do so with only a $\frac{1}{4}$ chance of detection.

Naively, one might suggest amplifying the security of the bit commital protocol by running it many times. This would allow us to realize a bit-commital protocol that is indistinguishable from an ideal protocol. However, such a strategy turns out to be inefficient, since even with an ideal scheme, the rejection probability will still only be 1/2. Thus, n (where $(\frac{7}{8})^n < \frac{1}{2}$) iterations of our protocol, using the simple commital scheme, will prove more efficacious than a single iteration of our protocol, with an arbitrarily amplified commital scheme. For efficiency, it makes sense to use a few very cheap iterations rather a single very expensive one. Metaphorically, we have a weakest link phenomenon: There is no point paying to make some links of a chain very strong if a single other link in this chain will still be weak.

5 References

[Bl] Blum, "How to Prove a Theorem So No One Else Can Claim It", Zero Knowledge Proofs, ICM 1986.

[BGKW] Ben-Or, Goldwasser, Kilian, and Wigderson, "Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions," Proceedings of STOC 1988.

[BCC] Brassard, Gilles, David Chaum, and Claude Crépeau, "Minimum Disclosure Proofs of Knowledge," JCSS, Oct. 1988.

[FS] Fiat, and Shamir, " How to Prove Yourself: Practical Solutions to Identification and Signature Problems", CRYPTO 86.

[GQ] Guillou, and Quisquater, "A Paradoxical Identity Based Signature Scheme Resulting from Zero Knowledge", CRYPTO 88.

[GMR] Goldwasser, Micali, and Rackoff, "The Knowledge Complexity of Interactive Proofs", SIAM J. of Comp., Feb. 1989.

[GMW] Goldreich, Micali, and Wigderson, "Proofs that Yield Nothing But the Validity of the Assertion", Proceedings of FOCS 1986.

[IY] Impagliazzo, Russell and Moti Yung, "Direct Minimum Knowledge Computations," CRYPTO 87.

[KMO] Kilian, Micali, and Ostrovsky, "Efficient Zero Knowledge Proofs with Bounded Interaction," Proceedings of FOCS 89.

[LO] Lagarias, and Odlyzko, "Solving Low Density Subset Sum Problems", Proceedings of FOCS 1983.

[MS] Micali and Shamir, "An Improvement of the Fiat-Shamir Identification and Signature Scheme", CRYPTO 88.

[Sh] A. Shamir, "A Zero-Knowledge Proof for Knapsacks", presented at a workshop on Probabilistic Algorithms, Marseille (March 1986).