Design and Security Analysis of Anonymous Group Identification Protocols

Chan H. Lee[1], Xiaotie Deng[1], and Huafei Zhu[2]

[1] Department of Computer Science,
City University of Hong Kong
{cschlee, csdeng}@cityu.edu.hk

[2] Zhejiang Univ. Zhejiang-Inst-Information Science and Engineering,
Department of Information and Electronics Engineering,
Hangzhou 310027 Peoples R. China
zhuhf@isee.zju.edu.cn

Abstract. Two provably secure group identification schemes are presented in this report: 1) we extend De Santis, Crescenzo and Persiano's (SCP) anonymous group identification scheme to the discrete logarithm based case; then we provide a 3-move anonymous group identification scheme, which is more efficient than that presented in [SCPM, CDS], with the help of this basic scheme; 2) we also extend the original De Santis, Crescenzo and Persiano anonymous group identification scheme to the general case where each user holds public key which is chosen by herself independently. The communication cost for one round execution of the protocol is $2mk$, where k is bit length of public key n and m is the number of users in the group.

1 Introduction

Anonymous group identification scheme is a method that allows a member of a group, say Alice convinces a verifier, say Bob that she is a member of the group without revealing any information about her identity in the group. A full and general solution to the problem of anonymous group identification has been presented in [SCPM] based on the closure properties of statistical zero knowledge languages under monotone logic formula composition. Later, Cramer, Damgard and Schoenmakers suggest a new general approach for proofs of partial knowledge independently [CDS]. The structure of an anonymous group identification protocol based on the techniques presented in [SCPM] (or [CDS], if we restrict CDS's scheme to the discrete logarithm based case), can be stated as follows [CM]: Let p be a large prime and G be a cyclic sub-group of Z_p^* with order q. Let g be a generator of G. The system public key is (p, q, g) available to all group users. Each user U_i in the group chooses $x_i \in z_q$ at random and computes $y_i := g^{x_i}$ mod$p$. The public key is y_i and the secret key is x_i. The set $S := \{y_1, \cdots, y_m\}$ contains the information of all legal users. A prover, say Alice wants to prove her knowledge of y_i to a verifier, say Bob without revealing any information of the index i. To authenticate membership of the group anonymously

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 188–198, 2002.

(without loss of generality, we assume the prover knows the secret information of y_1), the protocol is performed as follows:

- The prover computes $t_1 = g^{r_1}$, $t_2 = g^{s_2}{y_2}^{c_2}$, $\cdots$, $t_m = g^{s_m}{y_m}^{c_m}$, where $r_1 \in z_q$, $s_2, \cdots, s_m \in z_q$ and $c_2, \cdots, c_m \in z_q$ are chosen uniformly at random. Then she sends $t_1, t_2, \cdots, t_m$ to the verifier.
- The verifier chooses $b \in z_q$ uniformly at random and send it to the prover.
- The prover computes $c_1 = b \oplus c_2 \oplus \cdots \oplus c_m$, $s_1 = (r_1 - x_1 c_1) \text{mod } q$, then sends $(s_1, s_2, \cdots, s_m)$ and $(c_1, c_2, \cdots, c_m)$ to the verifier.
- The verifier tests whether $b = c_1 \oplus c_2 \oplus \cdots \oplus c_m$ and $t_1 = g^{s_1}{y_1}^{c_1}$, $\cdots$, $t_m = g^{s_m}{y_m}^{c_m}$. If all conditions are valid then the verifier accepts. Otherwise it rejects.

It is clear that the computational complexity is about $2m$-exponent computations for the prover (or the verifier) while the communication costs is about $3mn$-bit, with probability $1/2^n$ rejecting an unqualified group user, where m is the number of users, n is the bit length of the security parameter p ($p = 2q + 1$, p, q are two large primes).

Since the anonymity property often constrains the communication to heavily depend on the number of users in the group, communication cost is one of the most important measures for efficiency considerations. Motivated by the communication efficiency consideration, De Santis, Crescenzo and Persiano have developed elegant protocols minimizing the communication involved in the schemes. The proof of security is based on the difficulty of factoring problem. We sketch their basic protocol below (see [SCP] for more details):

- Initialization: Let n be a security parameter, uniformly choosing n-bit primes $p \equiv q \equiv 3 \text{ mod} 4$ and setting $x = pq$.
- For $i = 1, \cdots, m$, uniformly choosing $w_i \in Z_x^*$ and computing $y_i = {w_i}^2$ mod$x$; Setting $pk_i = (x, y_i)$ and $sk_i = w_i$ to user U_i;

To show the membership of the group anonymously, one round of protocol is performed as follows.

- The prover chooses $r \in Z_x^*$ and $c_1, \cdots, c_m \in \{0, 1\}$ uniformly at random, then sends $u = r^2 {y_1}^{c_1} \cdots {y_m}^{c_m}$ mod$x$ to the verifier;
- The verifier chooses a bit $b \in \{0, 1\}$ uniformly at random and sends it to the prover.
- If $b = c_1 \oplus \cdots \oplus c_m$, the prover sets $d_j = c_j$, for $j = 1, \cdots, m$ and $s = r$; If $b \neq c_1 \oplus \cdots \oplus c_m$, then sets $s = rw_i^{1-2d_i}$ and $d_i = 1 - c_i$. Finally the prover sends $(s, d_1, \cdots, d_m)$ to the verifier.
- The verifier checks whether $b = d_1 \oplus \cdots \oplus d_m$ and $u = s^2 {y_1}^{d_1} \cdots {y_m}^{d_m}$ mod$x$. If the both conditions hold, the verifier outputs *Accept*; else it outputs *Reject*;

It is clear that one round execution of the protocol rejects an un-qualified user with probability $1/2$. The computational cost is $(\frac{m}{2} + 1)$ modular multiplications (modular x), on average while the communication cost is $(m+2n)$-bit. If the protocol runs n times, then the total computation cost is $(m+2)n$ modular multiplications (modular p, we use the same measurement of modular multiplication) and the communication cost is $(m+2n)n$-bit, where m is the number of users in the group and n is bit length of private key p ($|p|$=$|q|$=n, and $|x|$=$2n$).

1.1 A Simple Modification of SPC's Scheme

Since the original SPC's scheme is multiple-round identification protocol. It is desired if one is able to provide a standard 3-move scheme. A natural modification of multiple-round SPC's scheme to a 3-move anonymous identification protocol can be stated as follows.

- The prover chooses $r \in Z_x^*$ and $c_1, \cdots, c_m \in z_q$ uniformly at random, then sends $u = r^2 {y_1}^{c_1} \cdots {y_m}^{c_m}$ mod$x$ to the verifier;
- The verifier chooses $b \in z_q$ uniformly at random and sends it to the prover.
- If $b = c_1 \oplus \cdots \oplus c_m$, setting $d_j = c_j$, for $j = 1, \cdots, m$ and $s = r$; If $b \neq c_1 \oplus \cdots \oplus c_m$, setting $d_j = c_j$ $(j \neq i)$, $d_i = c_1 \oplus \cdots c_{i-1} \oplus b \oplus \cdots \oplus c_m$ and $s = rw_i^{c_i - d_i}$. Finally the prover sends $(s, d_1, \cdots, d_m)$ to the verifier.
- The verifier checks whether $b = d_1 \oplus \cdots \oplus d_m$ and $u = s^2 {y_1}^{d_1} \cdots {y_m}^{d_m}$ mod$x$. If the both conditions hold, it outputs *Accept*; else it outputs *Reject*;

Notice that any honest prover has to compute the value $w_i^{c_i - d_i}$ mod$x$ if $b \neq c_1 \oplus \cdots \oplus c_m$ (this event happens with overwhelming probability). It follows that the computational cost of the prover is $(m+1)$ exponential computation modular x. Equivalently, it needs $3(m+1)n$ modular multiplications (modular p), according to the well known square and multiply algorithm for modular exponentiation that requires, on average, $1.5n$ modular multiplications for an n-bit exponent. And the communication cost of this protocol is about $2(m+1)n$-bit.

1.2 Our Works

We are interested in the construction of a standard 3-round anonymous identification scheme (for example, a protocol is more efficient than the above simple modification of SCP's scheme). Our solution is follows: we first extend De Santis, Crescenzo and Persiano's (SCP) anonymous group identification scheme to the discrete logarithm based case; then we provide a 3-move anonymous group identification scheme, which is more efficient than that presented in [SCPM, CDS], with the help of this basic scheme. The computational cost of our 3-move scheme is $(m+1)$-exponent computations for a prover (or a verifier) and the communication cost is $(m+2)n$-bit, with probability $1/2^n$ rejecting an unqualified group member, where m is the number of users, n is the bit length of the security parameter p ($p = 2q+1$, p, q are two large primes).

We remark that to achieve the same security level as our 3-move identification scheme (with probability $1/2^n$ rejecting an unqualified group member), the original SCP's should be run n times independently. It follows that the total communication cost of running n times of the original SCP's anonymous identification scheme is $(m+2n)n$-bit while our 3-move protocol is $(m+2)n$-bit.

We point out that the computation cost of our 3-move protocol is $(m+1)$-exponent computations for a prover (or a verifier), that is, it needs $1.5(m+1)n$ modular p multiplication computation. However the computation cost of running n times of the original SCP's anonymous identification scheme is $(m+2)n$ modular multiplications (modular p, we use the same measurement of modular operation). It is clear that the computation complexity of our 3-move scheme costs slightly more than that of running n times of the original SCP's scheme. We believe that the computational inefficiency of our 3-move scheme is inevitable since the prover of the original SCP's scheme is required to reply a single bit challenge and hereby the exponential computation is NOT needed. This is the key difference between a one bit challenge scheme and n-bit challenge scheme.

We realize that the computation and communication complexity of the above mentioned protocols grow linearly with number of the member of the group. The growth could make anonymous authentication protocol impractical for very large dynamic groups. It is interesting problem if one is able to develop a new anonymous authentication protocol such that the computational and communication complexity to identify the membership of a user in a given group is constant, that is, both communication and computation complexity are independent on the number of users in the group, e.g., notable works of Boneh and Franklin [BF]. This is our further research topic.

2 Notions and Definitions

In this section, we introduce some useful notations presented in [SCP], then we provide security definition on anonymous group identification protocols.

Probabilistic algorithms. The notation $x \leftarrow \mathrm{S}$ denotes the random process of selecting element x from set S with uniform probability over S. The notation $y \leftarrow \mathrm{A(x)}$, where A is an probabilistic algorithm, denotes the random process of obtaining y when running algorithm A on input x. A probability space will be denoted by $\{R_1; \cdots; R_m : v\}$, where v denotes the value that the random value can assume, and $R_1, \cdots, R_m$ is a sequence of random processes generating value v. By $\mathrm{Prob}[R_1; \cdots; R_m : E]$ we denote the probability of event E, after execution of the random processes $R_1, \cdots, R_n$.

Interactive protocols. If A and B are two interactive probabilistic polynomial Turing machines, by pair (A, B) we denote an interactive protocol. By $\mathrm{TR}_{(\mathrm{A},\mathrm{B})}(x)$ we denote a set of transcripts of the interaction between A and B with common input x. The notation $t \leftarrow (\mathrm{A(y)}, \mathrm{B})(x)$ denotes the transcript t has been generated through an execution of the protocol (A, B), where x is a common input to A and B, and y is A' s private input. For any $t \in \mathrm{TR}_{(\mathrm{A},\mathrm{B})}(x)$, by $\mathrm{OUT}_{\mathrm{C}}(t)$ we denote the output of C, where $\mathrm{C} \in \{\mathrm{A}, \mathrm{B}\}$. We will say B ACCEPT if $\mathrm{OUT}_{\mathrm{B}}(t)$=ACCEPT.

An anonymous identification scheme consists of two phases: initial phase and identification phase. Each user holds with a pair of public and private key at the end of execution of the initial protocol. In identification phase, a user tries to convince the verifier of some statement which certifies her knowledge of secret key received in the initial phase. By INIT, we denote a initial protocol and by (P, V), we denote two-party protocol, where P is proof algorithm executed by A and V is a verification algorithm executed by the verifier B. With the help of these notations, we are able to present two equivalent security definitions which may be convenient for us to prove the security aspects of the proposed schemes.

Definition 1. It is convenient for us to define the security of an anonymous group identification scheme in the case that each user in the group shares common system public keys and one round execution of the protocol rejects an un-qualified user with probability 1/2. An anonymous identification scheme $\{\mathrm{INIT}, (\mathrm{P}, \mathrm{V})\}$ is secure if it satisfies:

- Correctness: For each user $U_i \in S$, $\mathrm{Prob}[(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V})(pk) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT] = 1$;
- Soundness: For any user $U_i \notin S$ and any probabilistic polynomial time algorithm, the advantage:P': $\mathrm{Prob}[(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}'(\phi), \mathrm{V})(pk) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT] - 1/2$ is negligible, where $\mathrm{P}'(\phi)$ indicates the input of private key is empty string;
- Anonymity: For any user $U_i, U_j \in S$ and any probabilistic polynomial time algorithm V', the probability space Π_1 and Π_2 are equal, where $\Pi_1 = [(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V}')(pk) : t]$ and $\Pi_2 = [(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_j), \mathrm{V}')(pk) : t]$.

Remark: There are several equivalent security definitions of a group identification scheme. We sketch the two notions, which are useful to define security aspects of a group identification scheme in different settings.

Definition 2. It is convenient for us to define the security in the case that each user in the group shares common system public keys and one round execution of the protocol is enough to authenticate her membership of the given group. We say an anonymous identification scheme $\{\mathrm{INIT}, (\mathrm{P}, \mathrm{V})\}$ is secure if it satisfies:

- Correctness: For each user $U_i \in S$, $\mathrm{Prob}[(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V})(pk) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT] = 1$;
- Soundness: For any user $U_i \notin S$ and any probabilistic polynomial time algorithm, the advantage of P': $\mathrm{Prob}[(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}'(\phi), \mathrm{V})(pk) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT]$ is negligible, where $\mathrm{P}'(\phi)$ indicates the input of private key is empty string;
- Anonymity: For any user $U_i, U_j \in S$ and any probabilistic polynomial time algorithm V', the probability space Π_1 and Π_2 are equal, where $\Pi_1 = [(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V}')(pk) : t]$ and $\Pi_2 = [(sk_1, \cdots, sk_m, pk) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_j), \mathrm{V}')(pk) : t]$.

Definition 3. It is convenient for us to define the security of an anonymous group identification scheme in the case that each user in the group holds with different public key and one round execution of the protocol rejects an un-qualified user with probability 1/2. We say an anonymous identification scheme $\{\mathrm{INIT}, (\mathrm{P}, \mathrm{V})\}$ is secure if it satisfies:

- Correctness: For each user $U_i \in S$, $\mathrm{Prob}[(pk_1, \cdots, pk_m, sk_i) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V})(pk_1, \cdots, pk_m) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT] = 1$;
- Soundness: For any user $U_i \notin S$, for any probabilistic polynomial time algorithm, the advantage of P': $\mathrm{Prob}[(pk_1, \cdots, pk_m) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}'(\phi), \mathrm{V})(pk_1, \cdots, pk_m) : \mathrm{OUT}_{\mathrm{V}}(t) = ACCEPT] - 1/2$ is negligible, where $\mathrm{P}'(\phi)$ indicates the input of private key is empty string;
- Anonymity: For arbitrary two different legitimate users $U_i, U_j \in S$ and any probabilistic polynomial time algorithm V', the probability space Π_1 and Π_2 are equal, where $\Pi_1 = [(pk_1, \cdots, pk_m, sk_i) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_i), \mathrm{V}')(pk_1, \cdots, pk_m) : t]$ and $\Pi_2 = [(pk_1, \cdots, pk_m, sk_j) \leftarrow \mathrm{INIT}(1^n); t \leftarrow (\mathrm{P}(sk_j), \mathrm{V}')(pk_1, \cdots, pk_m) : t]$.

Since a zero-knowledge proof system achieves un-linkability (i.e., separate identification transcripts can not be shown have been made by a single individual [SPH]). It is desirable if the anonymous identification scheme shares zero-knowledge property. The notion of zero knowledge proof was introduced by Goldwasser, Micali and Rackoff [GMR]. We sketch some useful notions below: Let R be a relationship over a language L. Let $x \in L$ and $w(x) := \{w : (x, w) \in R\}$ be a witness set such that the membership can be tested in polynomial time. A proof system (P, V) is called perfect zero-knowledge if for any probabilistic polynomial time Turing machine V', there exists a simulator $S_{V'}$ such that:

- $S_{V'}$ outputs $\perp$ with probability at most 1/2;
- And that conditioned on not outputting $\perp$, the simulator's output is distributed as the verifier's view in a real interaction with the prover.

A weak notion is called an honest verifier zero-knowledge: there is a simulator S that on input x produces conversations that are indistinguishable from the real conversations with input x between the honest prover and the honest verifier. Hence an honest verifier zero-knowledge protocol also implies the unlinkability in the sense of the computational indistinguishablity.

3 Basic Anonymous Identification Scheme

We first extend De Santis, Crescenzo and Persiano's (SCP) anonymous group identification scheme to the discrete logarithm based case.

3.1 Descriptions

Let p, q be two large primes such that $p - 1 = 2q$. Let G be a cyclic sub-group of Z_p^* with order q. Let g be a generator of G. The system public key is (p, q, g) available to all group users. Each user U_i in the group chooses $x_i \in z_q$ at random and computes $y_i := g^{x_i}$ mod$p$. The public key is y_i and the secret key is x_i. The set $S := \{y_1, \cdots, y_m\}$ contains all users. A prover, say Alice wants to prove her knowledge of y_i to a verifier, say Bob without revealing any information of the index i. One round of the protocol can be executed as follows:

- Alice chooses $r \in z_q$ and $c_i \in \{0, 1\}$ uniformly at random then sends the value $u := g^r\ y_1^{c_1} \cdots y_m^{c_m}$ to Bob.
- Bob chooses a bit $b \in \{0, 1\}$ uniformly at random and then sends it to Alice.
- Alice checks the validation whether $b = c_1 \oplus \cdots \oplus c_m$. If the equation is satisfied, then Alice sends $(r, c_1, \cdots, c_m)$ to Bob. Otherwise, Alice sets $c_j \leftarrow c_j$ and $c_i \leftarrow (1 - c_i)$ and $r \leftarrow (r + (2c_i - 1)x_i) \mod q$ and then sends $(r, c_1, \cdots, c_m)$ to Bob.
- Bob checks the validation of the two conditions $b = c_1 \oplus \cdots \oplus c_m$ and $u := g^r$ $y_1^{c_1} \cdots y_m^{c_m}$. If both conditions hold, he accepts; Otherwise he rejects.

The communication cost is $(m + 2n)$-bit, and the computation cost is 1-exponent computation plus $m/2$ multiplications over G for one round execution of the protocol. If the protocol runs n times then the computation cost is about $\frac{1}{2}(m + 3n)n$ modular multiplications (modular p) according to the well known square and multiply algorithm for modular exponentiation requires, on average $1.5n$ modular multiplications for an n-bit exponent.

3.2 Security Analysis

Since the protocol needs multiple-round interactions between the honest prover and the honest verifier, we want to show that the protocol is secure according to the definition 1.

Correctness. Correctness can be easily verified according to the definition 1.

Soundness. Suppose there exists unqualified user $U' \notin S$ and a probabilistic polynomial time proof algorithm P' with non-negligible advantage to make the honest verifier accept, then there exists a polynomial time algorithm P^* solves the discrete logarithm problem with non-negligible probability.

Proof. We need to show an algorithm that takes (g, y) as input and produces the $\mathrm{DL}_g(y)$ as output (given access to a subroutine that breaks the protocol). Now we are given an random element $y \in G$, the adversary (running U' and P' together) chooses a set of random elements $r_1, \cdots, r_m \in z_q$ and compute $y_i = y^{r_i}$. Then it chooses a random element $r \in Z_q$ and computes $u := g^r y_1^{c_1} \cdots y_m^{c_m}$. By assumption there is a probabilistic polynomial time proof algorithm P' with non-negligible advantage to make the honest verifier accept, it follows P' is able to output $(u; r'; (c'_1, \cdots, c'_m)) \leftarrow \mathrm{P}'(u; r; (c_1, \cdots, c_m))$ with non-negligible probability such that $c_1 \oplus \cdots \oplus c_m \neq c'_1 \oplus \cdots \oplus c'_m$ and u: $= g^r\ y_1^{c_1} \cdots y_m^{c_m} = g^{r'} y_1^{c'_1} \cdots y_m^{c'_m}$. Denote $d_i = c'_i - c_i$ $(1 \leq i \leq m)$. Hence with non-negligible probability, the adversary obtains the equation: $g^{r-r'} = y_1^{d_1} \cdots y_m^{d_m}$. It follows that

$r - r' = (r_1 d_1 + \cdots r_m d_m)\mathrm{DLog}(y) \bmod q$. Hence the adversary is able to compute the discrete logarithm of any randomly given element y with non-negligible probability. We arrive at the contradiction of the hardness assumption of the discrete logarithm problem.

Anonymity. Anonymity follows from the fact that the distribution of the variable $r \leftarrow (r + (2c_i - 1)x_i) \mod q$ over z_q (of the user U_i) and the distribution of the variable $r' \leftarrow (r' + (2c_j - 1)x_j) \mod q$ over z_q (of the user U_j) are uniformly distributed if r and r' are chosen uniformly at random from z_q.

Perfect zero-knowledge. This basic scheme shares zero-knowledge property. We want to show that for any probabilistic polynomial time Turing machine V', there exists a simulator $S_{V'}$ such that: 1) $S_{V'}$ outputs $\perp$ with probability at most 1/2; 2) Conditioned on not outputting $\perp$, the simulator's output is distributed as the verifier's view in a real interaction with the prover. The simulator $S_{V'}$ can be constructed as follows:

- Setting the random tape of V': Let $poly(\cdot)$ be a polynomial bound running time of V'. The simulator $S_{V'}$ starts by uniformly selecting a random string $Random \in \{0,1\}^{poly(|q|)}$, to be used as the contents of the random tape of V';
- Simulating the first step of the prover: The simulator selects $b' \in \{0,1\}$, $r \in z_q$ and $c_i \in \{0,1\}$ uniformly at random such that $b' = c_1 \oplus \cdots \oplus c_m$, and computes the value $u := g^r y_1^{c_1} \cdots y_m^{c_m}$;
- Simulating the verifier's first step: The simulator initiates an execution of V' by placing x on V''s common input tape, *Random* on its random tape and b' on its incoming message tape. After polynomial number of steps of V', the simulator can read the outgoing message b;
- Simulating the prover's second step: If $b' = b$, then the simulator halts with output $(u, b, r, c_1, \cdots, c_m)$;
- Failure of the simulator: Otherwise, the simulator halts with output $\perp$.

Using the hypothesis that V' is polynomial time, it follows that so is the simulator. It is left to show that $S_{V'}$ outputs $\perp$ with probability at most 1/2 and that conditioned on not outputting $\perp$, the simulator's output is distributed as the verifier's view in a real interaction with the prover.

Notice that regardless of the value of b', the distribution of the message that $S_{V'}$ sends at the first move is the same as the distribution of the messages that the honest prover sends. And the random string *Random* on its random tape chosen by the simulator is uniformly distributed. The fact implies that V's reply in the move 2 is independent of the value of b'. Consequently, $S_{V'}$ outputs $\perp$ with probability at most 1/2.

We denote by $\mu(u, b, r, c_1, \cdots, c_m)$ the distribution of the transcripts between the honest prover and the V' while $\nu(u, b, r, c_1, \cdots, c_m)$ be the distribution of simulator $S_{V'}$'s distribution. Since for every fixed u and *Random*, the value b of the output of V' is uniquely determined and the simulation of the prover first step conditioned on not outputting $\perp$ is same as that of the honest prover, it follows the distribution of the simulator's output is identical to that of the verifier's view in a real interaction with the honest prover.

3.3 A Variations of Basic Anonymous Identification Scheme

We now are able to provide a variation of this basic scheme. The key generation scheme is the same as that in the basic scheme presented in the above section. To identify the membership of the group (without loss of generality, we assume that the prover knows the secret information of y_1), the protocol is performed as follows:

- The prover chooses $s, d_1, c_2, \cdots, c_m \in z_q$ uniformly at random, and computes $u = g^s y_1^{d_1} y_2^{c_2} \cdots y_m^{c_m}$ and then sends u to the verifier.
- The verifier chooses $b \in z_q$ uniformly at random and sends it to the prover.
- The prover computs $c_1 = b \oplus c_2 \cdots \oplus c_m$, and $r = s + (d_1 - c_1)x_1 \bmod q$. Then sends $(r, c_1, c_2, \cdots, c_m)$ to the verifier.
- If both conditions $b = c_1 \oplus c_2 \cdots \oplus c_m$ and $u = g^r y_1^{c_1} y_2^{c_2} \cdots y_m^{c_m}$ are satisfied, it accepts, Otherwise, it rejects.

3.4 Security Analysis

The proof of correctness, soundness and anonymity are the same as that presented in the basic scheme. To show the protocol is the honest verifier zero-knowledge proof system, we choose $b \in z_q$ at random, then we choose $c_1, \cdots, c_m \in z_q$ such that $b = c_1 \oplus c_2 \cdots \oplus c_m$. Finally we choose $r \in z_q$ uniformly at random and compute $u = g^s y_1^{d_1} y_2^{c_2} \cdots y_m^{c_m}$. The conversation is $(u, b, r, c_1, \cdots, c_m)$. Since b, r and $c_1, \cdots, c_m \in z_q$ are chosen uniformly at random, it follows the simulated conversation is indistinguishable from the real conversation.

The computational complexity is $(m+1)$-exponent computations for a prover (or a verifier), that is the total computation complexity is $1.5(m+1)n$ modular multiplications (modular p). And the communication cost is $(m+2)n$-bit, with the probability $1/2^n$ rejecting an unqualified group member. The facts imply that our 3-move scheme is more efficient than that presented in [SCPM, CDS].

4 Anonymous Group Identification Protocol with Independent Modular

In this section, we extend the original De Santis, Crescenzo and Persiano anonymous group identification scheme to the general case where each user holds public key which is chosen by herself independently. The communication cost for one round execution of the protocol is $2mk$, where k is bit length of public key n and m is the number of users in the group.

Let S be a set of elements $\{n_1, n_2, \cdots, n_m\}$. Each n_i is a product of two primes, P_i and Q_i such that $P_i \equiv Q_i \equiv 3 \bmod 4$ (Any integer n_i with this property is called Blum integer). A prover, say Alice wants to prove to a verifier, say

Bob that she knows the factors of some $n_j \in S$ without revealing any information of the index j. One round of the protocol can be executed as follows:

- Alice selects m elements $x_i \in Z^*_{n_i}$ at random, squares them to get $a_i = x_i^2 \bmod n_i$ ($a_i \neq a_j$ if $i \neq j$), then sends a_i to Bob ($i = 1, 2, \cdots m$);
- Bob randomly chooses a bit $b \in \{-1, 1\}$ and sends it to Alice.
- Alice sends Bob m square roots $z_1, z_2, \cdots, z_m$ of $a_1, a_2, \cdots, a_m$ such that $(\frac{z_1}{n_1})(\frac{z_2}{n_2}) \cdots (\frac{z_m}{n_m}){=}b$, where $(\frac{z}{n})$ is Jacobi symbol and $(z_1, z_2, \cdots, z_m)$ is either $(x_1, \cdots, x_{i-1}, x_i, \cdots, x_m)$ or $(x_1, \cdots x_{i-1}, y_i, \cdots, x_m)$.
- Bob checks validation of $z_i^2 \equiv a_i \bmod n_i$ ($i = 1, 2, \cdots, m$) and $(\frac{z_1}{N_1})(\frac{z_2}{N_2}) \cdots (\frac{z_m}{N_m}) = b$. If both tests passed, he accepts, otherwise he rejects.

Security analysis. To show the protocol is secure, the following result is needed, which can be found in [BSMP]:

- Fact 1: If a is a quadratic residue modulo n, where $P \equiv Q \equiv 3 \bmod 4$, then a has four square roots modulo N, denoted by $x, -x, y, -y$;
- Fact 2: If $x^2{=}y^2 \bmod N$ and if $x \neq \pm y \bmod N.$, where $P \equiv Q \equiv 3 \bmod 4$ and $N = PQ$, then $(\frac{x}{n}){=}(\frac{-x}{n})$ and $(\frac{x}{n}){=}-(\frac{y}{n})$.

Correctness. If Alice knows two factors of $n_i \in S$, then she can always make Bob accept.

Proof. Alice computes $(\frac{x_1}{n_1})(\frac{x_2}{n_2}) \cdots (\frac{x_m}{n_m})$ and compares it with b. If it is equal then sends $x_1, x_2, \cdots, x_m$ to Bob. Otherwise, Alice replaces x_i by y_i such that $x_i^2 \equiv y_i^2 \equiv a_i \bmod n_i$ and $(\frac{x_i}{n_i}){=}-(\frac{y_i}{n_i})$, then she sends $x_1, \cdots, y_i, \cdots, x_m$ to Bob. This is true if Alice knows two prime factors of $n_i \in S$. Hence Alice can always make Bob accept.

Soundness. Suppose there exists unqualified user $U' \notin S$ and a probabilistic polynomial time proof algorithm P' with non-negligible advantage to make the honest verifier accept, then there exists a polynomial time algorithm P^* factoring the Blum integer with non-negligible probability.

Proof. Given m legitimate users $n_1, \cdots, n_m$ in the group S, two factors of each n_i ($1 \leq i \leq m$) is not known by the U' since she is not a legitimate user in the group S. By protocol, U' must commit the values x_i ($x_i^2 \equiv a_i \bmod n_i$, $1 \leq i \leq m$) and then sends the commitments to the verifier at the first move. Since b is a random bit chosen by the verifier, by assumption P' is able to provide $(z_1, z_2, \cdots, z_m)$ which is pairs either $(x_1, \cdots, x_{i-1}, x_i, \cdots, x_m)$ or $(x_1, \cdots x_{i-1}, y_i, \cdots, x_m)$ for some i with non-negligible probability. The fact implies that there exists a polynomial time algorithm P^* (by running U' and P' together) factoring the Blum integer with non-negligible probability.

Anonymity. The distribution of transcript of any legitimate user is equal.

Proof. Any legitimate user, say Alice sends Bob random sequence $a_1, \cdots, a_m$ at first. After she receives challenge bit b, Alice sends the correspondent square roots sequence $z_1, \cdots, z_m$ such that $(\frac{z_1}{n_1})(\frac{z_2}{n_2}) \cdots (\frac{z_m}{n_m}){=}b$ to Bob. Notice that the distribution of $\{(\frac{z_1}{n_1}), (\frac{z_2}{n_2}), \cdots, (\frac{z_m}{n_m})\}$ is uniform over $\{1, -1\}^m$ if the distribution of random variant $(z_1, z_2, \cdots, z_m)$ is uniform. Hence the distribution of transcript of any legitimate user is uniform. It follows the protocol achieves anonymity property.

5 Conclusion

We have remarked that the computation and communication complexity of the above mentioned protocols grow linearly with number of the member of the group and the growth could make anonymous authentication protocol impractical for very large dynamic groups. It is desired if one is able to develop a new anonymous authentication protocol such that both computational and communication complexity to identify the membership of a user in a given group is constant, as notable works of Boneh and Franklin's. This is our further research topic.

Acknowledgements

Research is partially supported by research grants from City University (Project No. 7001023,7001232,7100133).

References

BF. D. Boneh, and M. Franklin. Anonymous authentication with subset queries. In proceedings of the 6th ACM conference on Computer and Communications Security, pp. 113–119.

BP. N. Braic and B. Pfitzmann. Collision free accumulators and fail-stop signature scheme without trees. Eurocrypt'97, 480-494, 1997.

BSMP. M. Blum, A. De Santis, S. Micali, and G. Persiano, Non-Interactive Zero Knowledge, SIAM Journal on Computing, vol. 19, n. 6, December 1991, pp. 1084-1118.

CDS. R. Cramer, I. Damgaard, B. Schoenmakers: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols, Proceedings of CRYPTO '94, Santa Barbara Ca., Springer Verlag LNCS, vol. 839, pp. 174-187.

SPH. S. Schechter, T. Parnell, A. Hartemink. Anonymous Authentication of Membership in Dynamic Group. Financial Cryptography'99. Springer-Verlag, 1999, 184-195.

SCP. A. De Santis, and G. Di Crescenzo and G. Persiano. Commmunication-efficient Anonymous Group Identification. With An extended abstract appears in the Proc. of the Fifth ACM Conference on Computer and Communications Security. 1998, page 73-82.

SCPM. A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung, On Monotone Formula Closure of SZK, Proceedings of 35th IEEE Symposium on Foundations of Computer Science (FOCS '94), Santa Fe, New Mexico, USA, November 20-22, 1994, pp. 454-465.