

Research Reports on Mathematical and Computing Sciences

Secret Handshake with Multiple Groups

Naoyuki Yamashita and Keisuke Tanaka

November 2006, C-229

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES **C**: Computer Science

Secret Handshake with Multiple groups

Naoyuki Yamashita and Keisuke Tanaka*

Dept. of Mathematical and Computing Sciences
Tokyo Institute of Technology
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{yamashi1, keisuke}@is.titech.ac.jp

November 14, 2006

Abstract

A privacy-preserving authentication model called secret handshake was introduced by Balfanz, Durfee, Shankar, Smetters, Staddon, and Wong [1]. It allows two members of a same group to authenticate themselves secretly to the other whether they belong to a same group or not, in the sense that each party reveals his affiliation to the other only if the other party is also a same group member. The previous works focus on the models where each participant authenticates himself as a member of one group. In this paper, we consider a secret handshake model with multiple groups. In our model, two users authenticate themselves to the other if and only if each one's memberships of multiple groups are equal. We call this model *secret handshake with multiple groups*. We also construct its concrete scheme. Our scheme can easily deal with the change of memberships. Even if a member is added to a new group, or deleted from the one that he belongs to, it is not necessary to change the memberships for the other groups that he belongs to.

Keywords: Secret Handshake, Authentication, Privacy, Anonymity.

1 Introduction

1.1 Background

A privacy-preserving authentication model called secret handshake was introduced by Balfanz, Durfee, Shankar, Smetters, Staddon, and Wong [1]. It allows two members of a same group to authenticate themselves secretly to the other whether they belong to a same group or not, in the sense that each party reveals his affiliation to the other only if the other party is also a group member.

For example, a CIA agent Alice might want to authenticate herself to Bob, but only if Bob is also a CIA agent. Moreover, if Bob is not a CIA agent, the protocol should not help Bob in determining whether Alice is a CIA agent or not.

The work of [1] constructed a secret handshake scheme secure under the bilinear Diffie-Hellman assumption in the random oracle model. Castelluccia, Jarecki, and Tsudik [2] constructed a secret handshake scheme, which is secure under the computational Diffie-Hellman (CDH) assumption in the random oracle model, based on an ID-based-like encryption scheme.

The above schemes [1, 2] are based on one-time credentials to achieve the unlinkability, which means that the attacker cannot specify the user even if he is a participant of the scheme. Without one-time credentials, Xu and Yung [6] constructed the scheme with the unlinkability.

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 16092206.

Furthermore, Tsudik and Xu [5] proposed a multi-party secret handshake model. In this model, with a single run of the protocol, any number of members can authenticate themselves to the others if and only if all of them belong to a same group. They also modified the unlinkability for the multi-party secret handshake model, and constructed a concrete scheme satisfying this property.

1.2 Our contribution

The previous works focus on the models where each participant authenticates himself as a member of one group. In this paper, we consider a secret handshake model with multiple groups, where two users authenticate themselves to the other if and only if each one's memberships of the groups are equal. We call this model *secret handshake with multiple groups*.

For example, assume that a CIA agent Alice is investigating a gang secretly, and she wants to meet a CIA colleague who is investigating the same gang, too. She meets a suspicious person, Bob. She wants to assure that he is both a CIA agent and an investigator of the gang. If he is not a CIA member or an investigator of the gang, she does not want to tell him either that she is a CIA member or that she is a investigator of the gang.

We also construct a concrete scheme for secret handshake with multiple groups. Our scheme can easily deal with the change of memberships. Even if a member is added to a new group, or is deleted from the one that he belongs to, it is not necessary to change his other memberships.

1.3 Organization

In Section 2, we propose a model of secret handshake with multiple groups. In Section 3, we present a concrete scheme of this model. In Section 4, we prove that our scheme satisfies the security requirement under the CDH assumption in the random oracle model.

2 Definition of secret handshake with multiple groups

In this section, we propose a model of secret handshake with multiple groups.

2.1 Model

We adapt the definition of [2] to secret handshake with multiple groups.

In our model, there is a group authority GA for each group. A scheme for secret handshake with multiple groups consists of four algorithms Setup, CreateGroup, AddMember, and Handshake.

- Setup takes as input the security parameter k and generates the public parameters params common to all subsequently generated groups.
- CreateGroup is a key generation algorithm executed by GA on input of params , and outputs the group public key G and the GA's private key x_G .
- AddMember is a protocol executed between a user and the group authority GA of G . The private input is GA's private key x_G . The common inputs are params , G , and the user's identity ID of size regulated by params . Then, the user gets a trapdoor t for the above ID . The user keeps the trapdoor secret.
- Handshake is the authentication protocol, i.e. the secret handshake protocol itself. It is executed between players A and B on public inputs ID_A , ID_B , and params . The private input of A is $(t_1, \dots, t_n, G_1, \dots, G_n)$, and the private input of B is $(t'_1, \dots, t'_{n'}, G'_1, \dots, G'_{n'})$. It outputs *accept* or *reject*.

We note that in all secret handshake schemes discussed in this paper the output of the Handshake protocol can be extended to include an authenticated session key along with the “accept” decision.

2.2 Basic security properties

We also adapt the definition of [2] to secret handshake with multiple groups.

A secret handshake scheme with multiple groups must have the following security properties: the completeness, the impersonator resistance, and the detector resistance. In some cases, the unlinkability is preferable.

2.2.1 Completeness

Assume that honest users A, B belonging to the same groups, that is, A belongs to G_1, \dots, G_n and B belongs to $G'_1, \dots, G'_{n'}$, then $n = n'$ and $\{G_1, \dots, G_n\} = \{G'_1, \dots, G'_{n'}\}$. If A and B run Handshake with valid trapdoors for their IDs and group public keys, then both parties output “accept”.

2.2.2 Impersonator resistance

The impersonator resistance property is violated if an adversary \mathcal{A} authenticates himself as a member of G_1, \dots, G_n to an honest user V when \mathcal{A} does not belong to at least one of G_1, \dots, G_n . Formally, we say that a secret handshake scheme is *impersonator resistant* if every polynomially bounded adversary \mathcal{A} has negligible probability of winning in the following game, for any string ID_V :

1. We execute $\text{params} \leftarrow \text{Setup}(1^k)$, and $(G_i, x_i) \leftarrow \text{CreateGroup}(\text{params})$ for $i = 1, \dots, n$.
2. The adversary \mathcal{A} , on input (G_1, \dots, G_n, ID_V) , invokes the **AddMember** algorithm on any groups any times. That is, for any bitstring ID and a public key of group G , \mathcal{A} can get a trapdoor t for ID and G .
3. When \mathcal{A} is ready for the challenge, \mathcal{A} is allowed to choose up to $n - 1$ groups and receives these GAs’ private keys x_i .
4. \mathcal{A} announces a new $ID_{\mathcal{A}}$, which is not included in any of the above queries.
5. \mathcal{A} interacts with the honest player V with the **Handshake** protocol. Common inputs are $(ID_{\mathcal{A}}, ID_V)$, and V ’s private inputs are (G_i, t_i) for $i = 1, \dots, n$.

We say that \mathcal{A} *wins* if V outputs “accept” in the above game.

We note that the above property is rather weak, and that stronger versions of this property are possible. Namely, the attacker is allowed to run the protocol several times against V , and is able to invoke the additional **AddMember** algorithm after each attempt. Also, the attacker is allowed to ask for trapdoors on additional $ID \neq ID_{\mathcal{A}}$ strings during the challenge protocol with V . We use a simple definition here. It can be shown that our scheme remains secure under these stronger notions.

2.2.3 Detector resistance

An adversary \mathcal{A} violates the detector resistance property if \mathcal{A} can decide whether some honest party V is a member of some groups G_1, \dots, G_n when \mathcal{A} does not belong to at least one of G_1, \dots, G_n . Formally, we say that a secret handshake scheme is *detector resistant* if there exists a probabilistic polynomial-time algorithm SIM , such that any polynomially bounded adversary \mathcal{A}

cannot distinguish between the following two games with probability non-negligibly higher than $1/2$, for any target ID string ID_V :

Steps 1 to 4 proceed as in the definition of the impersonator resistance, that is, on input ID_V and a randomly generated G_1, \dots, G_n , \mathcal{A} queries GA on adaptively chosen ID . \mathcal{A} is allowed to choose up to $n - 1$ groups to receive the GAs' private keys x_i . \mathcal{A} announces a new $ID_{\mathcal{A}}$, which is not included in any of the above queries.

- 5-1. In game 1, \mathcal{A} interacts with the honest player V with the Handshake protocol. the common input is $(ID_{\mathcal{A}}, ID_V)$, and V 's private inputs are (G_i, t_i) for $i = 1, \dots, n$.
- 5-2. In game 2, \mathcal{A} interacts with SIM on the common input $(ID_{\mathcal{A}}, ID_V)$.
6. \mathcal{A} can query GA on additional strings $ID \neq ID_{\mathcal{A}}$.
7. \mathcal{A} outputs "1" or "2", making a judgement on which game he saw.

Similarly to the impersonator resistance, stronger notions of the detector resistance are possible. In particular, the adversary should be able to trigger several executions of the handshake protocol with player V , and he should be able to replace these instances with those executed with the legitimate owner of the $ID_{\mathcal{A}}$ identity. We use the above weak notion for simplicity, but our scheme satisfies these stronger notions.

2.2.4 Unlinkability

A potentially desirable property is the unlinkability, which extends privacy protection for group members by requiring that instances of the handshake protocol performed by the same party cannot be efficiently linked. This property is violated if after an adversary \mathcal{A} interacts with the honest player V with the Handshake protocol, \mathcal{A} can determine the other's ID when \mathcal{A} does not belong to at least one of G_1, \dots, G_n . Formally, we say that a secret handshake scheme is *unlinkable* if any polynomially bounded adversary \mathcal{A} cannot distinguish between the following two IDs with probability non-negligibly higher than $1/2$, for any string ID_{V_1}, ID_{V_2} :

Steps 1 to 4 proceed as in the definition of the impersonator resistance except for the inputs on \mathcal{A} . On input ID_{V_1}, ID_{V_2} and a randomly generated G_1, \dots, G_n , \mathcal{A} queries GA on adaptively chosen ID . \mathcal{A} is allowed to choose up to $n - 1$ groups to receive the GAs' private keys x_i . \mathcal{A} announces a new $ID_{\mathcal{A}}$, which is not included in any the above queries.

5. We choose $b \in \{1, 2\}$ randomly. \mathcal{A} interacts with the honest player V with the Handshake protocol. The common input is $(ID_{\mathcal{A}}, ID_{V_b})$, and V 's private inputs are (G_i, t_i) for $i = 1, \dots, n$.
6. The adversary \mathcal{A} can query GA on additional strings $ID \neq ID_{\mathcal{A}}$.
7. The adversary \mathcal{A} outputs "1" or "2", making a judgement on whom he interacted with.

3 Concrete scheme

In this section, we construct a concrete scheme for secret handshake with multiple groups. This four-round scheme satisfies the security properties under the CDH assumption in the random oracle model. This scheme can be considered as a variant of [2], based on the Schnorr signature scheme [4] and the ElGamal encryption scheme.

- **Initialize** picks the standard discrete logarithm parameters (p, q, g) of security parameter k , that is, primes p, q of size k , such that g is a generator of a subgroup in \mathbb{Z}_p^* of order q . **Initialize** also defines hash functions $H : \{0, 1\}^* \times \langle g \rangle \rightarrow \mathbb{Z}_q$ and $H' : \langle g \rangle \rightarrow \langle g \rangle$. The hash functions are modeled as random oracles.
- **CreateGroup** picks a random private key $x \in \mathbb{Z}_q$ and sets the public group key $G = g^x$. Each G_i can be represented as a string and we can sort G_i 's lexicographically.
- In **AddMember** on a public input (G, ID) , the **GA** picks $r \in \mathbb{Z}_q$ randomly, and computes $w = g^r$ and $t = xH(ID, w) + r \pmod q$. The user's outputs are the certificate w and the trapdoor t .
- **Handshake** proceeds as follows. Assume that A 's inputs are $ID, (G_i, w_i, t_i)$ for $i = 1, \dots, n$ and B 's inputs are $ID', (G'_j, w'_j, t'_j)$ for $j = 1, \dots, n'$ where G_i 's and G'_j 's are sorted lexicographically.

1. B sends $(ID', w'_1, \dots, w'_{n'})$ to A .
If $n \neq n'$, A outputs *reject*.
 A obtains $PK'_1 = w'_1 G_1^{H(ID', w'_1)}, \dots, PK'_{n'} = w'_{n'} G_n^{H(ID', w'_{n'})}$.
 A picks $m_a \leftarrow_R \langle g \rangle$.
 A picks $c \leftarrow_R \mathbb{Z}_q$ and computes $(c_1, c_2) = (g^c, m_a H'(PK'_1{}^c) \dots H'(PK'_{n'}{}^c))$.
2. A sends $(ID, w_1, \dots, w_n, c_1, c_2)$ to B .
 B obtains $PK_1 = w_1 G_1^{H(ID, w_1)}, \dots, PK_n = w_n G_n^{H(ID, w_n)}$.
 B picks $m_b \leftarrow_R \langle g \rangle$.
 B picks $c' \leftarrow_R \mathbb{Z}_q$ and computes $(c'_1, c'_2) = (g^{c'}, m_b H'(PK_1{}^{c'}) \dots H'(PK_n{}^{c'}))$.
 B computes $m = H'(c_1^{t_1})^{-1} \dots H'(c_1^{t_n})^{-1} c_2$ and $resp_b = H'(m)$.
3. B sends $(c'_1, c'_2, resp_b)$ to A .
If $resp_b \neq H'(m_a)$, A outputs *reject*.
Otherwise, A computes
 $m' = H'(c_1^{t_1})^{-1} \dots H'(c_1^{t_n})^{-1} c'_2$ and $resp_a = H'(m')$.
4. A sends $resp_a$ to B .
If $resp_a \neq H'(m_b)$, B outputs *reject*.
Otherwise B outputs *accept*.

3.1 Discussion

Clearly, our scheme does not satisfy the unlinkability. However, by the following extension, this property can be satisfied. In steps 1 and 2 of **Handshake**, one can run the protocol using multiple IDs and certificates. Then, the other picks multiple challenge messages and encrypts them with the IDs and certificates. After receiving these ciphertexts, he computes the plaintexts and responds hashed plaintexts. The other authenticates him if he can decrypt one of them. If he can decrypt none of them, he rejects. We prove this property in section 4.3.

4 Security of our scheme

In this section, we prove that our scheme satisfies the security properties under the CDH assumption in the random oracle model.

It is clear that our scheme satisfies the correctness. That is, if honest users belonging to the same groups run **Handshake** with valid trapdoors for their IDs and group public keys, then **Handshake** outputs “accept”.

4.1 Impersonator resistance

Theorem 1. *Our scheme is impersonator resistant under the CDH assumption in the random oracle model.*

Proof. We prove this security property by reduction. By using the adversary that attacks this property with non-negligible probability ϵ , we construct the adversary \mathcal{A}^* that solves the CDH problem with non-negligible probability.

The adversary \mathcal{A} attacks against an honest user V identified by ID_V who is a member of n groups. We use \mathcal{A} as the A and V as the B in the definition of the Handshake. It is not necessary to consider the other case.

On the input of the Diffie-Hellman challenge (g, g^a, g^d) , \mathcal{A}^* chooses $l \in \{1, \dots, n\}$ and sets $G_l = g^a$. We assume $l = n$ without loss of generality. \mathcal{A} chooses $x_1, \dots, x_{n-1} \leftarrow \mathbb{Z}_q^*$ randomly and computes $G_1 = g^{x_1}, \dots, G_{n-1} = g^{x_{n-1}}$. \mathcal{A}^* inputs (G_1, \dots, G_n, ID_V) to \mathcal{A} . Let $x_n = a$.

When \mathcal{A} queries ID to the AddMember algorithm of the k -th group, \mathcal{A}^* simulates as follows. If $k \neq n$, \mathcal{A}^* actually computes the Schnorr signature on string ID under the GA's secret key x_k and returns a pair (w, t) such that $w = g^r$ and $t = x_k H(ID, w) + r$. If $k = n$, \mathcal{A}^* simulates the Schnorr signature. \mathcal{A}^* picks $i, t \leftarrow_R \mathbb{Z}_q^*$ randomly, computes $w = g^t (G_n^i)^{-1}$, sets $H(ID, w) = i$, and sends (t, w) to \mathcal{A} . Since this pair satisfies the verification equation and i, t are picked at random, \mathcal{A}^* simulates the random oracles.

When \mathcal{A} announces that he is ready for the impersonation challenge against V , \mathcal{A} is allowed to choose up to $n - 1$ groups to receive the GAs' private keys x_i . Since \mathcal{A} that receives $n - 1$ private keys has the largest probability to success the attack, we can assume that \mathcal{A} chooses $n - 1$ groups. If \mathcal{A} chooses n -th group to receive the private key, \mathcal{A}^* halts. Otherwise, \mathcal{A}^* passes x_1, \dots, x_{n-1} . Then \mathcal{A} passes $(ID_{\mathcal{A}}, w_1, \dots, w_n)$ to \mathcal{A}^* . In the step 3 of Handshake algorithm, \mathcal{A}^* sets $c_1 = g^d$, $c_2 \leftarrow_R \langle g \rangle$ and passes (c_1, c_2) to \mathcal{A} . Assume that, for each k , $w_k = g^{r_k}$ and t_k is the trapdoor of \mathcal{A} for the k -th group. In the random oracle model, the probability that \mathcal{A} makes the correct answer $resp = H'(m)$ without querying m to H' such that $c_2 = m H'(c_1^{t_1}) \cdots H'(c_1^{t_n})$ is negligible. Thus, in order to compute m , \mathcal{A} has to query $c_1^{t_i}$ to H' for $i = 1, \dots, n$. Therefore, \mathcal{A} can exponentiate a random element c_1 to exponent t_1, \dots, t_n .

In the above argument, after receiving Schnorr signatures (t_i, w_i) on \mathcal{A} 's choice, \mathcal{A} will compute $(w_n, c_1^{t_n})$ such that $w_n = g^{r_n}$ and $t_n = x_n H(ID_{\mathcal{A}}, w_n) + r_n$ for some $r_n, ID_{\mathcal{A}}$.

We apply the forking lemma by Pointcheval and Stern [3]. Let TM be a probabilistic polynomial time Turing machine, given only the public data as input. Let $(m, \sigma_1, h, \sigma_2)$ be a signature in the forking lemma where h is the hash value of (m, σ_1) and σ_2 just depends on σ_1 , the message m , and h . The forking lemma shows that if TM can find, with non-negligible probability, a valid signature $(m, \sigma_1, h, \sigma_2)$, then, with non-negligible probability, a replay of this machine, with the same random tape and a different oracle, outputs two signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$. The forking lemma used in the security proof of the Schnorr signature scheme shows that if there exists an attacker that breaks the existential unforgeability under an adaptive chosen message attack with non-negligible probability, then the discrete logarithm in subgroups can be solved in polynomial time. This means that if two conversations with an adversary and different random oracles produce the same message and signature, then $x = DL_g(G)$ can be computed.

In our proof, we reduce the the successful attack not to computing discrete logarithms, but to computing the CDH problem g^{ad} . We can consider $(ID_{\mathcal{A}}, w_n, H(ID_{\mathcal{A}}, w_n), c_1^{t_n})$ as a tuple $(m, \sigma_1, h, \sigma_2)$ in the forking lemma. Recall that \mathcal{A}^* has set $c_1 = g^d$. In the first conversation, \mathcal{A} receives $H(ID_{\mathcal{A}}, w_n) = j$ and computes

$$\begin{aligned} c_1^{t_n} &= c_1^{x_n H'(ID_{\mathcal{A}}, w_n) + r_n} \\ &= g^{d(a_j + r_n)}. \end{aligned}$$

In the second conversation, \mathcal{A} receives $H(ID_{\mathcal{A}}, w_n) = j'$ and computes

$$c_1^{t'_n} = g^{d(aj'+r_n)}.$$

After these two conversations, \mathcal{A}^* can compute $g^{ad} = (c_1^{t_n}/c_1^{t'_n})^{(j-j')^{-1}}$.

Since x_1, \dots, x_n are chosen randomly, the probability that \mathcal{A} does not choose n -th group to receive the GA's private key x_n is $1/n$. If the probability of \mathcal{A} to break the impersonator resistance is ϵ , the probability that \mathcal{A} wins the game twice with the same $(ID_{\mathcal{A}}, w_n)$ is at least ϵ^2/n^2 . Then, \mathcal{A}^* can return the answer to the CDH challenge with probability $\epsilon^2/(n^2q_h)$ where q_h is the number of queries that \mathcal{A} makes to the hash function H' . If the success probability ϵ is non-negligible, \mathcal{A} is an efficient algorithm, and hence the number of queries q_h is polynomial, then the probability that \mathcal{A}^* can return the correct answer of CDH is non-negligible. \square \square

4.2 Detector resistance

Theorem 2. *Our scheme is detector resistant under the CDH assumption in the random oracle model.*

Proof. We prove this security property by a similar way as in the proof of the impersonator resistance. By using an adversary that attacks this property with probability $1/2 + \epsilon$, we construct the adversary \mathcal{A}^* that solves the CDH problem with non-negligible probability.

The adversary \mathcal{A} attacks against an honest user V identified by ID_V , that is a member of n groups. We use \mathcal{A} as the A and V as the B in the definition of the Handshake. It is not necessary to consider the other case.

\mathcal{A}^* sets the values $ID_V, x_1, \dots, x_{n-1}, G_1, \dots, G_n$, simulates AddMember algorithm, and sets the challenge response (c_1, c_2) as in the proof of the impersonator resistance.

If \mathcal{A} distinguishes a conversation with V from a conversation with SIM, he reveals the information $w_1, \dots, w_n, (c_1, c_2)$ of the groups from the response from V . w_1, \dots, w_n and c_1 are random values and independent of the group public key. Since the probability that \mathcal{A} reveals the information without querying m such that $c_2 = mH'(c_1^{t_1}) \cdots H'(c_1^{t_n})$ is negligible, \mathcal{A} has to query $c^{(i)} = c_1^{t_i}$ to H' for $i = 1, \dots, n$. Therefore, \mathcal{A} can exponentiate a random element c_1 to exponent t_1, \dots, t_n . In the above argument, after receiving Schnorr signatures (t_i, w_i) on \mathcal{A} 's choice, \mathcal{A} will compute $(w_n, c_1^{t_n})$ such that $w_n = g^{r_n}$ and $t_n = x_n H(ID_{\mathcal{A}}, w_n) + r_n$ for some $r_n, ID_{\mathcal{A}}$.

Again, by applying the forking lemma, \mathcal{A}^* can compute g^{ad} with non-negligible probability as in the proof of impersonator resistance. \square \square

4.3 Unlinkability

In this section, we prove that the modified scheme satisfies the unlinkability. Therefore, we consider the modification as discussed in section 3.1. In step 1 and 2 of the Handshake, one can run the protocol using multiple IDs and certificates. Then, the other picks multiple challenge messages and encrypts them with the IDs and certificates. After receiving these ciphertexts, he computes the plaintexts and responds hashed plaintexts. The other authenticates him if he can decrypt one of them. If he can decrypt none of them, he rejects.

Theorem 3. *The modified scheme is unlinkable under the CDH assumption in the random oracle model.*

Proof. We prove this security property by a similar way as in the proof of the impersonator resistance. By using an adversary that attacks this property with probability $1/2 + \epsilon$, we construct the adversary \mathcal{A}^* that solves the CDH problem with non-negligible probability.

The adversary \mathcal{A} attacks against an honest user V identified by ID_{V_1} , that is a member of n groups. We use \mathcal{A} as the A and V as the B in the definition of the Handshake. It is not necessary to consider the other case.

On the input of the Diffie-Hellman challenge (g, g^a, g^d) , \mathcal{A}^* chooses $l \in \{1, \dots, n\}$ and sets $G_l = g^a$. We assume $l = n$ without loss of the generality. \mathcal{A} chooses $x_1, \dots, x_{n-1} \leftarrow \mathbb{Z}_q^*$, $ID_{V_2} \leftarrow \{0, 1\}^*$ randomly and computes $G_1 = g^{x_1}, \dots, G_{n-1} = g^{x_{n-1}}$. \mathcal{A}^* inputs $(G_1, \dots, G_n, ID_{V_1}, ID_{V_2})$ to \mathcal{A} . Let $x_n = a$.

When \mathcal{A} queries ID to the `AddMember` algorithm of the k -th group, \mathcal{A}^* simulates as follows. If $k \neq n$, \mathcal{A}^* computes the Schnorr signature on string ID under the GA's secret key x_k and returns a pair (w, t) such that $w = g^r$ and $t = x_k H(ID, w) + r$. If $k = n$, \mathcal{A}^* simulates the Schnorr signature. \mathcal{A}^* picks $i, t \leftarrow_R \mathbb{Z}_q^*$ randomly, computes $w = g^t (G_n^i)^{-1}$, sets $H(ID, w) = i$, and sends (t, w) to \mathcal{A} . Since this pair satisfies the verification equation and i, t are picked at random, \mathcal{A}^* can simulate the random oracles.

When \mathcal{A} announces that he is ready for the unlinkability challenge against V , \mathcal{A} is allowed to choose up to $n - 1$ groups to receive the GAs' private keys x_i . Since \mathcal{A} that receives $n - 1$ private keys has the largest probability to success the attack, we can assume that \mathcal{A} chooses $n - 1$ groups. If \mathcal{A} chooses n -th group to receive the private key, \mathcal{A}^* halts. Otherwise, \mathcal{A}^* passes two tuples of $(ID_{V_1}, w_1^{V_1}, \dots, w_n^{V_1})$ and $(ID_{V_2}, w_1^{V_2}, \dots, w_n^{V_2})$ to \mathcal{A} . Then \mathcal{A} passes two tuples of $(ID_{\mathcal{A}}, w_1, \dots, w_n)$ and $(ID'_{\mathcal{A}}, w'_1, \dots, w'_n)$ to \mathcal{A}^* . In the step 3 of Handshake algorithm, \mathcal{A}^* sets $r \leftarrow_R \{1, 2\}$, $c_{1r} = g^d$ and $c_{2r}, c_{1(3-r)}, c_{2(3-r)} \leftarrow_R \langle g \rangle$ and passes these $(c_{1r}, c_{2r}), (c_{1(3-r)}, c_{2(3-r)})$ to \mathcal{A} . Assume that, for each k , $w_k = g^{r^k}, w'_k = g^{r'^k}$ and t_k or t'_k is the trapdoor of \mathcal{A} for the k -th group. If \mathcal{A} can distinguish a conversation with ID_{V_1} from a conversation with ID_{V_2} , he reveals the information $w_1, \dots, w_n, w'_1, \dots, w'_n, (c_{11}, c_{21}), (c_{12}, c_{22})$ of the IDs from the response from V . $w_1, \dots, w_n, w'_1, \dots, w'_n, c_{11}$, and c_{12} are random values and independent from the IDs. The probability that \mathcal{A} reveals the information of IDs without querying m such that $c_{21} = mH'(c_{11}^{t_1}) \cdots H'(c_{11}^{t_n})$ or m' such that $c_{22} = m'H'(c_{12}^{t'_1}) \cdots H'(c_{12}^{t'_n})$ is negligible, \mathcal{A} has to query $c_{11}^{t_1}$ to H' for $i = 1, \dots, n$ or $c_{12}^{t'_1}$ to H' for $i = 1, \dots, n$. Therefore, \mathcal{A} can exponentiate a random element c_{11} to exponent t_1, \dots, t_n or c_{22} to exponent t'_1, \dots, t'_n .

In the above argument, after receiving signatures (t_i, w_i) on ID_i on \mathcal{A} 's choice, \mathcal{A} will compute a message $ID_{\mathcal{A}}$ and its signature $(w_n, c_{11}^{t_n})$ such that $w_n = g^{r^n}$ and $t_n = x_n H(ID_{\mathcal{A}}, w_n) + r_n$ or $(w'_n, c_{12}^{t'_n})$ such that $w'_n = g^{r'^n}$ and $t'_n = x_n H(ID'_{\mathcal{A}}, w'_n) + r'_n$ for some $r_n, ID_{\mathcal{A}}$.

Again, by applying the forking lemma, \mathcal{A}^* can compute g^{ad} with non-negligible probability as in the proof of impersonator resistance. □

5 Conclusion

We proposed a model for *secret handshake with multiple groups*, and constructed its concrete scheme. Our scheme can easily deal with one's change of membership. Even if a member is added to a new group, or deleted from the one that he belongs to, it is not necessary to change his other memberships.

It might be interesting to consider other extensional variations of secret handshake.

References

- [1] BALFANZ, D., DURFEE, G., SHANKAR, N., SMETTERS, D. K., STADDON, J., AND WONG, H.-C. Secret handshakes from pairing-based key agreements. In *IEEE Symposium on Security and Privacy* (2003), pp. 180–196.

- [2] CASTELLUCCIA, C., JARECKI, S., AND TSUDI, G. Secret Handshakes from CA-Oblivious Encryption. In *ASIACRYPT* (Jeju Island, Korea, December 2004), P. J. Lee, Ed., vol. 3329 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 293–307.
- [3] POINTCHEVAL, D., AND STERN, J. Security proofs for signature schemes. In *EUROCRYPT* (Saragossa, Spain, May 1996), U. Maurer, Ed., vol. 1070 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 387–398.
- [4] SCHNORR, C.-P. Efficient Identification and Signatures for Smart Cards. In *CRYPTO* (Santa Barbara, California, USA, August 1989), G. Brassard, Ed., vol. 435 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 239–252.
- [5] TSUDI, G., AND XU, S. A flexible framework for secret handshakes. In *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing* (New York, NY, USA, 2005), ACM Press, pp. 39–39.
- [6] XU, S., AND YUNG, M. k -anonymous secret handshakes with reusable credentials. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security* (New York, NY, USA, 2004), ACM Press, pp. 158–167.