SCIS 2003 The 2003 Symposium on Cryptography and Information Security
Hamamatsu,Japan, Jan.26-29,2003
The Institute of Electronics,
Information and Communication Engineers

Anonymous Authentication Protocol for Dynamic Groups with Power-Limited Devices

Jongseong Kim * jskim224@icu.ac.kr
Soogil Choi * sooguri@icu.ac.kr
Kwangjo Kim * kkj@icu.ac.kr
Colin Boyd † boyd@isrc.qut.edu.au

Abstract— We propose an anonymous authentication protocol that not only allows much lower computational complexity for practical use but also meets requirements of dynamic groups, especially with power-limited devices. Our contribution is to provide the strict analysis of security based on the framework of provable security. Our protocol consists of a group manager, a verification center, and m group members. In the protocol, the verification center that acts like a TTP does not possess any information needed to identify group members. Similar to the verification center, the group manager also cannot identify the group member without the verification center's help. In particular, our protocol is suitable for *English* auction or open bidding systems where the change of participating entities in a group occurs frequently and, at the same time, the processing must be executed within short time period.

Keywords: Anonymous authentication, Dynamic group, English auction, Hash chain, Open bidding

1 Introduction

Authentication and *dynamic group management* are indispensable components in English auction and open procurement, which are very popular business areas in E-commerce. In these systems, a group member wants to participate in the group activity without revealing her identity except when honor is awarded to herself as a winner. This is the basic problem of *anonymous authentication*.

As the previous works, witness-indistinguishability [6, 7] based or Zero-Knowledge [1, 8, 9] based anonymous authentication has been achieved [2, 11, 15, 16]. Up to now, most of previous works have been tried to reduce computation and communication complexity in their protocols. Yet no schemes are practical enough to be used in environment with power-limited devices such as smart cards or mobile devices. Another important concern is that managing a group dynamically is a crucial task for a group manager since every group is alive and has a variant life cycle.

Therefore, we focus ourselves on 2 points: efficient management of dynamic groups and low complexity of computation. In the real world, E-commerce activities using dynamic groups *occur frequently* for a day or even for a minute. In this case, there are some important activities like (re) forming groups, managing membership, and selecting a winner. As a viewpoint of the group manager, managing above activities dynamically and precisely becomes a critical task since the life cycle of groups lasts not long but rather short. Furthermore, members *should make decision quickly and bid timely* to win against competitors. Therefore, it is reasonable that 1) groups are managed efficiently and 2) all computations are done in a cheap way.

1.1 Related Work

Over the years, several papers [2, 4, 5] have attempted to study on the anonymous authentication. The group signature scheme introduced by Chaum and van Heyst [5] allows members of a group to sign messages on behalf of a group such that the resulting signature does not reveal their identities. But the public key of a group depends on the size of the group[1].

In [4], Camenisch and Stadler presented an efficient solution of the key-increasing problem. They proposed a signature of the knowledge of the discrete logarithm, which is basically a modification of Schnorr signature [17].

Boneh and Franklin [2] proposed anonymous authentication schemes based on proof of knowledge for the l-th root of modulo n and the RSA scheme. In [16], an anonymous authentication protocol using public key set of all group members was introduced. As pointed out by the authors, Verifiably Common Secret (VCS) space grows linearly with the number of the group members.

Recently, Omoto and Miyaji [14] proposed an efficient public auction protocol and Lee *et al.* [12] improved [14] to solve fairness of the winning bidder by publicly posting the winner's identity.

1.2 Outline

The remainder of this paper is organized as follows. Section 2 describes some primitives used in our pro-

* International Research center for Information Security (IRIS), Information and Communications Univ. (ICU), 58-4 Hwaam-dong, Yuseong-gu, Daejeon, 305-732, Korea
† Information Security Research Center, School of Data Communications, Queensland University of Technology, Australia

[1] In general, it is called the key increasing problem

posal along with cryptographic notions, and Section 3 gives the precise definition that should be met by an anonymous authentication protocol. Section 4 presents our protocol, Section 5 discusses its security and efficiency. We end with concluding remarks in Section 6.

2 Description of Primitives

This section describes the basis of Diffie-Hellman scheme and the universal one-way hash function (UOWHF)-used for Lamport one-time password scheme [10].

2.1 Diffie-Hellman Scheme

The mathematical tool commonly used for devising key agreement protocols is the *computational Diffie-Hellman* (CDH) *problem*: given a cyclic group G of prime order n, a generator g of G, and elements $g^x, g^y \in_{\mathcal{R}} G$ (where $x, y \in [1, n-1]$), then find g^{xy}. ($x \in_{\mathcal{R}} S$ denotes that x is chosen randomly from the set S.)

Definition 2.1 *A* Diffie-Hellman scheme (DHS) [3] *is a pair of polynomial time algorithms, $\langle \mathcal{G}_{\mathsf{DH}}, \mathcal{C} \rangle$. On input 1^k, $\mathcal{G}_{\mathsf{DH}}$ generates a triple of global parameters (p, q, g) where p and q are primes such that $q|(p-1)$, and g is an element of order q in $\boldsymbol{Z}_p^*$. $\mathcal{C}$ outputs $g^x \bmod p$.*

An adversary $\mathcal{A}$ of the DHS is a probabilistic polynomial time (PPT) algorithm which takes as input a parameter set (p, q, g) generated using $\mathcal{G}_{\mathsf{DH}}$, and a pair (g^{R_1}, g^{R_2}) for $R_1, R_2 \in_{\mathcal{R}} \mathbf{Z}_q^*$. The output of $\mathcal{A}$ is an element $\alpha (= g^{R_1 R_2})$ of $\mathbf{Z}_p^*$.

Definition 2.2 *We say a* DHS *is secure when the success probability* $\mathsf{Succ}^{\mathsf{DH}}(\mathcal{A})$ *defined by*

$$\mathsf{Succ}^{\mathsf{DH}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathcal{D}}\big((p,q,g),(g^{R_1},g^{R_2})\big) = \alpha \;\middle|\; \begin{array}{l} (p,q,g) \leftarrow \mathcal{G}_{\mathsf{DH}}(1^k); \\ R_1, R_2 \leftarrow \boldsymbol{Z}_q^*; \\ \alpha = g^{R_1 R_2} \end{array} \right]$$

is negligible for every $\mathcal{A}$ given access to Diffie-Hellman oracle $\mathcal{D}$.

2.2 UOWHF-based Lamport Scheme

We replace OWF in the Lamport scheme by universal one-way hash functions (UOWHF) [13] and extend to the UOWHF-based Lamport scheme.

Definition 2.3 *Let $\mathcal{H}$ be a collection of functions such that for all $h \in_{\mathcal{R}} \mathcal{H}$, $h : \{0,1\}^{n_{1_k}} \mapsto \{0,1\}^{n_{0_k}}$ for any two constants $n_{1_k} \geq n_{0_k}$. Let $\boldsymbol{H}$ be a family of UOWHFs $\mathcal{H}$. A* UOWHF-based Lamport scheme (ULS) *is a deterministic polynomial time algorithm $\mathcal{H}^{(\cdot)}(\cdot)$ for $\mathcal{H} \in_{\mathcal{R}} \boldsymbol{H}$. Given a secret s, UOWHF $\mathcal{H}$ chosen uniformly at random is used to define the increasing sequence: $s_{n-1} = \mathcal{H}(s), s_{n-2} = \mathcal{H}^2(s), \ldots, s_0 = \mathcal{H}^n(s)$. To authenticate a user for the i-th commitment, $0 \leq i \leq n-1$, a secret is defined to be $s_i = \mathcal{H}^{n-i}(s)$.*

$\mathcal{A}$ of the ULS is a PPT algorithm which has access to an oracle that computes ULS for any i-th commitment under a randomly chosen secret s. The output of $\mathcal{A}$ is a secret $s_i = \mathcal{H}^{n-i}(s)$.

Definition 2.4 *The success probability in forging a* ULS *of $\mathcal{A}$, given access to a* ULS*ing oracle $\mathcal{L}$, is*

$$\mathsf{Succ}^{\mathsf{ULS}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathcal{L}}(\mathcal{H}, s_i) = t \;\middle|\; \begin{array}{l} s \leftarrow \{0,1\}^k; \\ \mathcal{H} \leftarrow \boldsymbol{H}; \\ s_i = \mathcal{H}^{n-i}(s); \\ t_i = \mathcal{H}^{n-i}(t); \\ s_i = t_i, s \neq t \end{array} \right].$$

The probability is taken over the choices of the ULS *algorithm, and of $\mathcal{A}$.*

Each user selects a secret s as a seed and chooses uniformly at random a UOWHF $\mathcal{H} \in_{\mathcal{R}} \mathbf{H}$ to calculate the password chain. A user gives initial password $s_0 = \mathcal{H}^n(s)$ to the verifier. For the i-th commitment, $0 \leq i \leq n-1$, the user sends the i-th value $s_i = \mathcal{H}^{n-t}(s)$ where t must greater than that of previous commitment step.

Definition 2.5 *$\mathcal{A}$ of the* ULS *$(\tau, q_{\mathcal{L}}, \varepsilon)$*-breaks *a* ULS *scheme if $\mathcal{A}$ runs in time at most τ, makes at most $q_{\mathcal{L}}$ queries to the* ULS*ing (or UOWHF) oracle $\mathcal{L}$, and $\mathsf{Succ}^{\mathsf{ULS}}(\mathcal{A}) \geq \varepsilon$.*

Definition 2.6 *A* ULS *is a $(\tau, q_{\mathcal{L}}, \varepsilon)$*-secure ULS *if no adversary $(\tau, q_{\mathcal{L}}, \varepsilon)$-breaks it.*

3 Definitions of Security

3.1 Anonymous Authentication Protocols

To manage a group dynamically, the following 5 requirements are essential.

R1. *Security*: Only members of a group can be authenticated.

R2. *Anonymity*: Not even GM can know the identity of a member.

R3. *Unlinkability*: Transactions cannot be identified that who makes and sends.

R4. *Formationability*: GM can efficiently build and maintain new groups.

R5. *Maintenanceability*: GM can easily add to or remove members from the group.

Definition 3.1 *An* anonymous authentication (AA) *protocol is a quadruple $P_{\mathsf{AA}} = (\mathcal{G}_{\mathsf{AA}}, \mathcal{R}_{\mathsf{AA}}, \mathcal{C}_{\mathsf{AA}}, \mathcal{I}_{\mathsf{AA}})$ of PPT computable algorithms involving U_i, GM, and VC defined by the followings:*

- *$\mathcal{G}_{\mathsf{AA}}$ runs the* DHS *to create a secret session key K_{GV}, generates a random number set and chooses a nonce T which offers the randomness of transactions.*
- *$\mathcal{R}_{\mathsf{AA}}$ specifies how U_i registers to GM. $\mathcal{R}_{\mathsf{AA}}(1^k, U_i, R_i, T, \mathcal{H}, s_{U_i})$ outputs "*Accept*" or "*Reject*" according to the verification of VC, where 1^k denotes the security parameter, R_i denotes a random number assigned by GM for U_i, $\mathcal{H}$ denotes a UOWHF, and s_{U_i} denotes a secret seed of U_i.*

- $\mathcal{C}_{\mathsf{AA}}$ *specifies how* U_i *commits.* $\mathcal{C}_{\mathsf{AA}}(1^k, U_i, \mathsf{INFO}_{\mathsf{U_i}}, \mathsf{t}, \mathcal{H})$ *outputs "*OK*" if the verification of* VC *is valid.* $\mathsf{INFO}_{\mathsf{U_i}}$ *denotes any information that* U_i *may hold, and* t *means an arbitrary increased number for a commitment.*

- $\mathcal{I}_{\mathsf{AA}}$ *specifies how* GM *and* VC *can identify the qualified* U_i. $\mathcal{I}_{\mathsf{AA}}(1^k, R_i, t, \mathcal{H})$ *outputs an identity of a specific* U_i.

3.2 Anonymous Authentication Security

Definition 3.2 P_{AA} *is a* secure AA *(*s-AA*) protocol* $P_{s\text{-}\mathsf{AA}}$ *if the requirements* ***R1*** *and* ***R2*** *hold.*

R1. Security.

- *For all large enough* k*, for constants* n, i *and for all input* $s \in \{0,1\}^k$ *we have that*

$$\Pr\left[\mathcal{H}^i\left(\mathcal{H}^{n-i}(s)\right) \neq \mathcal{H}^n(s)\right]$$

is negligible (in k*).*

- *For any PPT* $\mathcal{A}$*, for input* k*,* i*, and* $s \in \{0,1\}^k$*, we have that*

$$\Pr\left[\begin{array}{l|l} \mathcal{A}^{\mathcal{L}}(\mathcal{H}, s_i) = t & \begin{array}{l} s_i = \mathcal{H}^{n-i}(s); \\ t_i = \mathcal{H}^{n-i}(t); \\ t_i = s_i; \\ s \neq t \end{array} \end{array}\right] < \epsilon(k)$$

where the probability is taken over random choices and all $\mathcal{H} \in_{\mathcal{R}}$ ***H*** *and the random choice of* $\mathcal{A}$.

R2. Anonymity.

- *For any PPT* $\mathcal{A}$ *with any set of group members* $\{U_i | 1 \leq i \leq m\}$*:*

$$\left|\Pr[\mathcal{A}(U_i, ...) = 1] - \Pr[\mathcal{A}(U_j, ...) = 1]\right| < \epsilon(k)$$

where $i \neq j$ *and* $\epsilon(k)$ *is a negligible function of* k.

We denote as $\mathsf{Succ}^{s\text{-}\mathsf{AA}}(\mathcal{A})$ the success probability that a PPT $\mathcal{A}$ violates the s-AA protocol. Consider the additional requirements such as unlinkability, formationability, and maintainability.

Definition 3.3 *Say that an* AA *protocol* P_{AA} *is a* robust AA *(*r-AA*) protocol* $P_{r\text{-}\mathsf{AA}}$ *if all of the 5 requirements hold.*

We denote as $\mathsf{Succ}^{r\text{-}\mathsf{AA}}(\mathcal{A})$ the success probability that a PPT $\mathcal{A}$ violates the r-AA protocol.

4 A Robust Anonymous Authentication Protocol

First, we give a brief outline of roles of each participant. Next, the detail description of the protocol follows.

4.1 Notations

For any message msg and a shared secret key K_{AB} between one participant A and the other participant B, we denote symmetric key encryption by $E(\mathsf{msg})_{\mathsf{K_{AB}}}$, asymmetric key encryption by $E_p(\mathsf{msg})$.

4.2 Protocol Participants

The roles of each entity are as follows:

Group Manager, GM
Primarily, GM manages groups. In addition, it runs DHS to generate K_{GV} with VC and assigns a random number R_i to U_i. A secret database for keeping member's identity and the assigned random number must be maintained.

Verification Center, VC
The major responsibility is to verify if the entity is valid or not. VC takes part in generating K_{GV} with GM. VC also must keep each member's hash value $\mathcal{H}^n(s_i)$.

Group member, U_i
To participate in the current group, U_i must send his identity to GM. He can commit himself many times using his chained hash values.

4.3 An r-AA Protocol

4.3.1 Preparation.

GM executes the sub-protocol $\mathcal{G}_{\mathsf{AA}}$ as follows:

1. GM chooses a group $G = \langle g \rangle$ of prime order q in which the CDH assumption holds and prepares a UOWHF $\mathcal{H} \in_{\mathcal{R}} \mathbf{H}$.
2. GM computes $g^{x_{GM}}$, signs on and sends it to VC. Also, VC does the same work. A shared secret key K_{GV} is created.
3. GM generates a random number set $(R_1, \ldots, R_n)$, encrypts it with K_{GV}, and sends it to VC.
4. On receiving the value, VC decrypts it using K_{GV} and keeps it secret in order.

4.3.2 Registration.

To register U_i, the sub-protocol $\mathcal{R}_{\mathsf{AA}}$ acts as follows only once:

1. U_i sends his identity to GM.
2. First, GM chooses R_i, encrypts it using K_{GV}. Second, GM encrypts it and a nonce T using a public key of U_i and sends it to U_i.
3. On receiving, U_i extracts $E(R_i)_{K_{GV}}$, encrypts it and $\mathcal{H}^n(s_{U_i})$ with the public key of VC and sends it to VC.
4. On receiving, VC decrypts it and verifies if $R_i \in (R_1, \ldots, R_n)$ is valid. If valid, he sends the message Accept to U_i and keeps $\langle R_i, \mathcal{H}^n(s_{U_i}) \rangle$ secret.

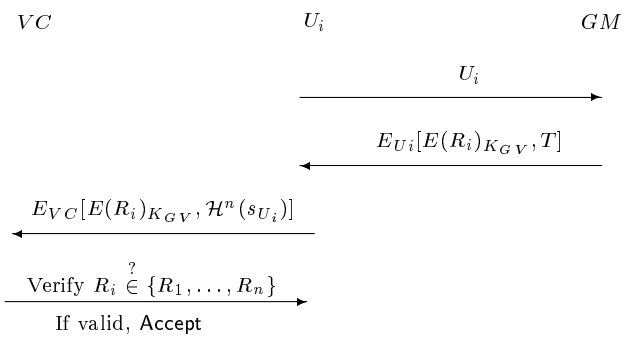

Figure 1: Registration using $\mathcal{R}_{\mathsf{AA}}$.

4.3.3 Commitment(s).

U_i can commit whenever he wants using the sub-protocol $\mathcal{C}_{\mathsf{AA}}$ as Figure 2.

1. At first, U_i determines the index t, retrieves the hash chain value $\mathcal{H}^{n-t}(s_{U_i})$ by the index t. Also, encrypts $\mathsf{INFO}_{\mathsf{U_i}}$ with GM's public key. U_i sends $[\mathcal{H}^{n-t}(s_{U_i}), t, E_{GM}(\mathsf{INFO}_{\mathsf{U_i}})]$ to GM.
2. When U_i commits, GM requests VC to check the validity of U_i by sending $[\mathcal{H}^{n-t}(s_{U_i}), t]$. (It is possible for GM to verify the validity of U_i by himself.[2])
3. On receiving, VC verifies $\mathcal{H}^t[\mathcal{H}^{n-t}(s_{U_i})] \stackrel{?}{=} \mathcal{H}^n(s_{U_i})$. If valid, then he sends the message "OK" to GM.

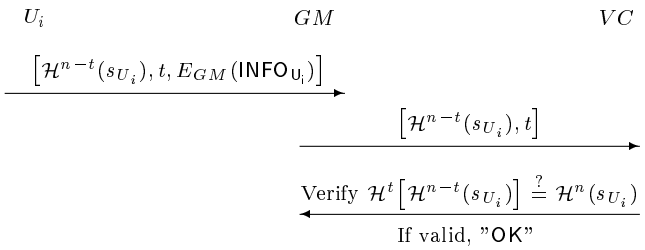

Figure 2: Commitment using $\mathcal{C}_{\mathsf{AA}}$.

4.3.4 Identification.

When an expiration date is over, GM and VC can identify the qualified U_i easily by using the sub-protocol $\mathcal{I}_{\mathsf{AA}}$ as shown in Figure 3.

1. GM sends $[\mathcal{H}^{n-t}(s_{U_i}), t]$, which is the winning member's commitment, to VC.
2. On receiving, VC verifies $\mathcal{H}^t[\mathcal{H}^{n-t}(s_{U_i})] \stackrel{?}{=} \mathcal{H}^n(s_{U_i})$.
3. If valid, then VC sends R_i related in $\mathcal{H}^n(s_{U_i})$ to GM.

[2] Note that in order to reduce the time required for the verification of VC, once after the initial verification, GM may keep $\mathcal{H}^{n-t}(s_{U_i})$ submitted by each member to check the validity of the group members by himself.

4. GM declares U_i associated with the random number R_i as just the winning user and writes U_i on the bulletin board.

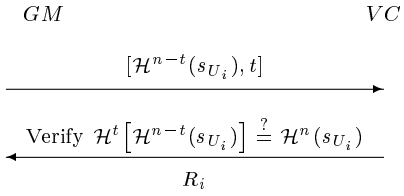

Figure 3: Identification using $\mathcal{I}_{\mathsf{AA}}$.

5 Analysis

5.1 Security Analysis

The following theorem shows the security of DHS in the protocol, especially in the sub-protocol $\mathcal{G}_{\mathsf{AA}}$.

Theorem 5.1 *Under the* CDH *assumption, for the sub-protocol* $\mathcal{G}_{\mathsf{AA}}$ *using* DHS*, let* $\mathcal{A}$ *be a PPT algorithm for the group* G *that makes at most* $q_{\mathcal{D}}$ CDH*-oracle queries. If* $x_{GM}, x_{VC} \in G$*, then* $\mathsf{Succ}^{\mathsf{DH}}(\mathcal{A}) \leq q_{\mathcal{D}}/|G|$.

Proof: We can easily construct from $\mathcal{A}$ a CDH adversary $\mathcal{F}$ that gets the success probability $\mathsf{Succ}^{\mathsf{DH}}_{\mathcal{R}_{\mathsf{AA}}}(\mathcal{F})$ in solving the DHP with in time $\bar{\tau}$.

$\mathcal{F}$ runs the protocol P_{AA} that provides random choices for $\mathcal{A}$, $\mathcal{G}_{\mathsf{AA}}$, and other participants and answers the queries made by $\mathcal{A}$ as follows. Based on the assumption that the CDH problem is hard, $\mathcal{A}$ cannot get any advantage in solving the CDH problem without having made a query the form $\mathcal{D}(R_1R_2)$, $R_1, R_2 \in_{\mathcal{R}} G$. At some point $\mathcal{A}$ makes a query to GM, $\mathcal{F}$ gets the value ω and relays $\mathcal{D}(\omega)$ to $\mathcal{A}$. $\mathcal{F}$ looks for ω in $\mathcal{D}$-list: $\mathcal{F}$ outputs 1 if ω is in the $\mathcal{D}$-list of queries made by $\mathcal{A}$, otherwise $\mathcal{F}$ outputs a random choice value.

The success probability of $\mathcal{F}$ is the probability that $\mathcal{A}$ made a query of the form $\mathcal{D}$ (R_1R_2) minus the probability that $\mathcal{A}$ made such a query by pure chance:

$$\begin{aligned}\mathsf{Succ}(\mathcal{F}) &= \Pr\left[\mathcal{A} \text{ makes query } (R_1R_2)\right] - \frac{q_{\mathcal{D}}}{|G|} \\ &\geq \mathsf{Succ}(\mathcal{A}) - \frac{q_{\mathcal{D}}}{|G|}.\end{aligned}$$

This completes the proof. □

Now we prove that the protocol P_{AA} satisfies two key requirements **R1** and **R2**.

Theorem 5.2 *Let* $\mathcal{A}$ *be an adversary that can get the probability* ϵ *in breaking the* s-AA *protocol* $P_{s\text{-}\mathsf{AA}}$ *within a time bound* τ*, after* $q_{\mathcal{D}}$ CDH *oracle queries and* $q_{\mathcal{L}}$ ULS*ing oracle queries. Then we have:*

$$\mathsf{Succ}^{s\text{-}\mathsf{AA}}(\mathcal{A}) \leq \frac{q_{\mathcal{D}}}{|G|} + \mathsf{Succ}^{\mathsf{DH}}(\tau', q_{\mathcal{D}}) + \frac{q_{\mathcal{L}}}{n \cdot 2^k} \cdot \mathsf{Succ}^{\mathsf{ULS}}(\tau'', \mathsf{q}_{\mathcal{L}})$$

where $\tau' \leq \tau + q_{\mathcal{D}} \cdot T_{exp}(k)$ *and* $\tau'' \leq \tau + q_{\mathcal{L}} \cdot T_{hash}(k)$*;* $T_{exp}(k)$ *is the time of computation required for an exponentiation modulo a* k*-bit number and* T_{hash} *is the time required for* ULS *hashing of a* k*-bit string.*

Proof: Any unqualified user cannot obtain R_i so that she does not pass the verification of VC. U_i just delivers the encrypted R_i to VC. Since U_i does not know the session key K_{GV}, he cannot read his R_i. This can prevent conspiracy attack with other group members.

So the proof of Theorem 5.2 now depends on the following lemmas.

Lemma 5.3 *Let* $\mathcal{F}$ *be a* CDH *adversary against the sub-protocol* $\mathcal{R}_{\mathsf{AA}}$ *on* P_{AA} *within time bound* τ'*, after* $q_{\mathcal{D}}$ CDH *oracle queries. Then the success probability of* $\mathcal{F}$ *that breaks the* CDH *problem is*

$$\mathsf{Succ}^{\mathsf{DH}}_{\mathcal{R}_{\mathsf{AA}}}(\mathcal{F}) \leq \frac{q_{\mathcal{D}}}{|G|} + \mathsf{Succ}^{\mathsf{DH}}(\bar{\tau})$$

where $\bar{\tau} \leq \tau' + q_{\mathcal{D}} \cdot T_{exp}(k)$*;* $T_{exp}(k)$ *is the time of computation required for an exponentiation modulo a* k*-bit number.*

Proof: The proof follows immediately from Theorem 5.1. The running time of $\mathcal{F}$ is the running time of $\mathcal{A}$ added to the time to process his exponentiation operation: $\bar{\tau} \leq \tau' + q_{\mathcal{D}} \cdot T_{exp}(k)$. □

Lemma 5.4 *Let* $\mathcal{F}$ *be a collision adversary against the sub-protocol* $\mathcal{R}_{\mathsf{AA}}$ *on* P_{AA} *within time bound* τ''*, after* $q_{\mathcal{L}}$ ULS*ing oracle queries. Then the success probability of* $\mathcal{F}$ *that finds any collision on UOWHF inputs is*

$$\mathsf{Succ}^{\mathsf{ULS}}_{\mathcal{R}_{\mathsf{AA}}}(\mathcal{F}) \leq \frac{q_{\mathcal{L}}}{n \cdot 2^k} \cdot \mathsf{Succ}^{\mathsf{ULS}}(\hat{\tau})$$

where $\hat{\tau} \leq \tau'' + q_{\mathcal{L}} \cdot T_{hash}(k)$*;* T_{hash} *is the time required for* ULS *hashing of a* k*-bit string.*

Proof: We only have to prove if $\mathcal{H}$ is a UOWHF, then the ULS is also a UOWHF. Now to prove the lemma, we show how $\mathcal{A}$ that finds collisions in a ULS can be transformed into a collision adversary $\mathcal{F}$ finds collisions in a UOWHF $\mathcal{H}$. This reduction can be quite made efficiently: the running time of $\mathcal{F}$ is basically the same as that of $\mathcal{A}$, and if $\mathcal{A}$ finds a collision with probability $\epsilon(k)$, then $\mathcal{F}$ finds a collision with probability about at least $\epsilon(k)/2^k$.

Let $s_{U_i} \in_{\mathcal{R}} \{0,1\}^k$ be an input of a user U_i to the ULS; for $1 \leq j \leq n$ give some n, define $s_{U_i}(j)$ be the first ℓ bits of the input to the j-th application of the UOWHF $\mathcal{H}$.

Consider the behavior of $\mathcal{A}$. Suppose its first message s_{U_i}[3] is formed as $s_{U_i,1}, \ldots, s_{U_i,n}$, and its second message s' that yields the collision is formed as $s'_1, \ldots, s'_n$. For this collision, we define δ be the smallest positive integer such that $s_{U_i}(n-\delta) \neq s'(n-\delta)$. The pair $\langle s_{U_i}(n-\delta), s'(n-\delta)\rangle$ will be the collision on $\mathcal{H}$ that $\mathcal{F}$ finds.

[3] It is called the target message in the literature of provable security.

The adversary $\mathcal{F}$ runs as follows. We let $\mathcal{A}$ choose uniformly its first message s_{U_i} at random. Then $\mathcal{F}$ guesses the value of δ at random. This guess will be correct with probability $1/2^k$. $\mathcal{F}$ now construct its target message as $\hat{s}(n-\delta)$, where $\hat{s}$ is, of course, an ℓ-bit string drawn uniformly from $\{0,1\}^k$ at random. The task of $\mathcal{F}$ is to generate a series of UOWHF values $\mathcal{H}^0, \ldots, \mathcal{H}^{n-1}$ such that has the correct distribution, and also that $s_{U_i}(n-\delta) = \hat{s}$. Once this is accomplished, $\mathcal{A}$ attempts to find a collision with s. If $\mathcal{A}$ succeeds, and if the guess at δ was right, this will yield a collision for $\mathcal{F}$. The probability that $\mathcal{F}$ outputs a collision is the probability that $\mathcal{A}$ succeeds in finding a collision multiplied to the probability to "correct guess" the i-th application of UOWHF $\mathcal{H}$:

$$\mathsf{Succ}^{\mathcal{H}}(\mathcal{F}) \geq \frac{q_{\mathcal{L}}}{n \cdot 2^k} \cdot \mathsf{Succ}^{\mathsf{ULS}}(\mathcal{A}, \mathsf{q}_{\mathcal{L}}).$$

Thus the collision adversary $\mathcal{F}$ runs in time $\tau'' + q_{\mathcal{L}} \cdot T_{hash}(k)$, and the result follows. □

That completes the proof of the theorem. In other words, the result shows that the first requirement **R1** holds on P_{AA}. □

Lemma 5.5 *Assuming that* $\mathcal{H}$ *is a UOWHF and the* CDH *problem is hard, for input parameters* $\mathcal{H}, t$*, and information* $\mathsf{INFO}_{\mathsf{U_i}}$ *and* $\mathsf{INFO}_{\mathsf{U_j}}$ *there is no efficient* $\mathcal{A}$ *such that can distinguish* U_i *and* U_j *with non-negligible probability.*

Proof: At the registration step, when GM assigns R_i to U_i, it knows the identity of U_i. At the commitment step, however, GM cannot know the identity of U_i since U_i submits $\langle \mathcal{H}^{n-t}(s_{U_i}), t, \mathsf{INFO}_{\mathsf{U_i}}\rangle$. GM has no information about a seed s_{U_i}. Similarly, VC cannot know the identity of U_i because he does not trace the link U_i with R_i.

To extract R_i during the registration step, $\mathcal{A}$ should defeat the security of DHS and ULS, which contradicts the proof of the above theorems. The result then follows. □

The theorem then follows from putting together the above equations. In the sequel, we deduced from P_{AA} the extended protocol $P_{s\text{-}\mathsf{AA}}$ ensuring its security and anonymity.

Theorem 5.6 *On* $P_{s\text{-}\mathsf{AA}}$*, there is no efficient adversary identify who makes which transactions.*

Proof: To achieve anonymity against GM, VC, and other group members, U_i should choose a new seed s_{U_i} per each session. However, VC realizes that s_{U_i} is the same as the seed of the previous session, he still does not know the owner of the seed. □

Theorem 5.7 *On* $P_{s\text{-}\mathsf{AA}}$*, the requirements* ***R4*** *and* ***R5*** *hold.*

Proof: At the preparation step, because only legitimated users who want to participate can be picked out, GM can make the group with no redundancy. To create a new group, GM only does the generation of a session key and a random number set. □

Corollary 5.8 *The above protocol* $P_{s\text{-AA}}$ *is* robust *if* $\mathcal{H}$ *in the* ULS *is a UOWHF and the* CDH *problem is hard for the* DHS *in the group* G*, which means that* $P_{s\text{-AA}}$ *is an* r-AA *protocol* $P_{r\text{-AA}}$.

5.2 Performance Analysis

We describe the features of our protocol compared to other schemes from the viewpoints of computation for all participants, which is shown in Table 1.

We denote by E modular exponentiation, by H an application of hash function, and by m the number of group members. We define n be a constant number on hash function. We assume that in public key cryptosystem, encrypting operation in general corresponds to two modular exponentiations.

Table 1: Computation Complexity

Procedure	[2]	[16]	[11]	[18]	Ours
Preparation	$(m+1)$E	mE	mE	$3m$E	4E
Registration	1E	8E	$(m+1)$E	4E	4E1H
Commitment	7E	4E2H	$(m+1)$E	2E1H	2E1H
Identification	$(m/2)$E	4E2H	2E	2E1H	1H

From the table, we see that the computation amount required in each procedure is drastically reduced compared with other schemes. This great decrease of whole computation results from avoiding using public-key cryptosystems such as public-key encryption and decryption or group signatures. Furthermore, we provide the strict analysis of security built on the provable security. As mentioned before, we present that our proposal meets two additional requirements (**R4** and **R5**).

6 Conclusion

We proposed an anonymous authentication protocol that satisfies main requirements for dynamic group communications: one is the efficient management of a dynamic group and the other is to require low computational complexity. In particular, we focused on the reduction of computation complexity only using hash function, which enables the efficient group management. We also attempted to provide the strict analysis of our proposal that determines whether crucial requirements are satisfied or not.

As our future work, we have to devote ourselves to providing complete security analysis. We expect that the low computational complexity implies that our protocol is suitable for *ad hoc* networks such as mobile communications.

References

[1] D. Boneh, "The decision Diffie-Hellman problem", *ANTS 1998*, LNCS 1423, pp. 48–63, 1998.

[2] D. Boneh and M. Franklin, "Anonymous authentication with subset queries", *ACM CCS 1999*, ACM Press, pp. 113–119, 1999.

[3] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis", *IMA ICCC1997*, LNCS 1355, pp. 30–45, 1997.

[4] J. Camenisch and M. Stadler, "Efficient group signature systems for large groups", *Crypto 1997*, LNCS 1294, pp. 410–424, 1997.

[5] D. Chaum and E. van Heyst, "Group signatures", *Eurocrypt 1991*, LNCS 547, pp. 257-265, 1991.

[6] U. Feige and A. Shamir, "Witness indistinguishability and witness hiding protocols", *Proc. of the 22nd STOC*, pp. 416–426, 1990.

[7] O. Goldreich, S. Goldwasser, and S. Micali, "Interleaved zero-knowledge in the public-key model", *ECCC*, TR99-024, 1999.

[8] O. Goldreich and H. Krawczyk, "On the compisition of zero-knowledge proof systems", *SIAM Journal on Computing*, 25(1):169–192, 1996.

[9] O. Goldreich, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM Journal on Computing*, 18:186-208, 1989.

[10] L. Lamport, "Password authentication with insecure communication", *Communications of the ACM*, 24(11), pp. 770–772, 1981.

[11] C.H. Lee, X. Deng, and H. Zhu, "Design and security analysis of anonymous group identification protocols", *PKC 2002*, LNCS 2274, pp. 188–198, 2002.

[KP98] J. Kilian and E. Petrank, "Identity Escrow", *Crypto'98*, LNCS 1462, pp. 169-185. 1998.

[12] B. Lee, K. Kim, and J. Ma, "Efficient public auction with one-time registration and public verifiability", *Indocrypt 2001*, LNCS 2247, pp. 162–174, 2001.

[13] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications", *Proc. of the 21th STOC*, pp. 33–43, 1989.

[14] K. Omote and A. Miyaji, "A practical English auction with one-time registration", *ACISP 2001*, LNCS 2119, pp. 221–234, 2001.

[15] A.D. Santis, G.D. Cresenzo, and G. Persiano, "Communication-effiecient anonymous group identification", *ACM CCS 1998*, pp. 73–82, 1998.

[16] S. Schechter, T. Parnell, and A. Hartemink, "Anonymous authentication of membership in dynamic groups", *Financial Cryptography 1999*, LNCS 1648, pp. 184–195, 1999.

[17] C.P. Schnorr, "Efficient identification and signatures for smart cards", *Crypto 1989*, LNCS 435, pp. 235–251, 1990.

[18] J. Kilian and E. Petrank, "Identity Escrow", *Crypto 1998*, LNCS 1462, pp. 169-185. 1998.