# A Quick Group Key Distribution Scheme with "Entity Revocation"

Jun Anzai[1] , Natsume Matsuzaki[1] , and Tsutomu Matsumoto[2]

[1] Advanced Mobile Telecommunications Security Technology Research Laboratories
3-20-8, Shinyokohama, Kohoku-ku, Yokohama, Kanagawa, 222-0033 Japan
{anzai, matuzaki }@amsl.co.jp
[2] Division of Artificial Environment and Systems, Yokohama National University,
79-5, Tokiwadai, Hodogaya, Yokohama,240-8501 Japan

**Abstract:** This paper proposes a group key distribution scheme with an "entity revocation", which renews a group key of all the entities except one (or more) specific entity (ies). In broadcast systems such as Pay-TV, Internet multicast and mobile telecommunication for a group, a manager should revoke a dishonest entity or an unauthorized terminal as soon as possible to protect the secrecy of the group communication. However, it takes a long time for the "entity revocation" on a large group, if the manager distributes a group key to each entity except the revoked one. A recently published paper proposed a group key distribution scheme in which the amount of transmission and the delay do not rely on the number of entities of the group, using a type of secret sharing technique. This paper devises a novel key distribution scheme with "entity revocation" that makes frequent key distribution a practical reality. This scheme uses a technique similar to "threshold cryptosystems" and the one-pass Diffie-Hellman key exchange scheme.

## 1 Introduction

It is required a secure and quick key distribution scheme suitable for such broadcast systems which dynamically change the compose of the group, as Pay-TV (broadcasting via satellite or via cable), Internet multicasts (push, streaming or conference systems, for example) and mobile telecommunication for a group (private mobile radio or taxi radio, for example).

In this paper, we focus on a group key distribution scheme for all the entities except one (or some) specific one(s), that is called "entity revocation" here. "Entity revocation" is an essential mechanism for a secure group communication, if we consider the situation when an unauthorized user might eavesdrop using a lost or stolen terminal of mobile telecommunications or when an entity that left off from a conference system has continued to hear the secret communication of the conference. Also "entity revocation" is necessary to prevent dishonest entities from enjoying a pay service like Pay-TV and pay Internet without paying a charge.

A Familiar method for "entity revocation", called "Familiar method" in this paper, is that a key distributor distributes a new group key to each entity except the revoked

K. Y. Lam, E. Okamoto and C. Xing (Eds.): ASIACRYPT'99, LNCS 1716, pp. 333-347, 2000.

entities, as encrypted form by a secret key of each entity. However, the amount of transmission and the delay become large when the group becomes large.

Papers [7] and [8] proposed a concept and a concrete scheme of a conference key distribution for secure digital mobile communications with low computational complexity. Also paper [2] proposed a method to expand Diffie-Hellman key exchange scheme so as to share a key among three or more entities through broadcast network. Since their schemes basically involve a key distribution for the other entities except the revoked one, the feature is the same as "Familiar method". So, one of our goal is to propose a scheme with "entity revocation", in which the amount of transmission and the delay do not rely on the group scale and analyze its security and performance.

On the other hand, a scheme proposed in paper [9] enables entities to share a key so that the amount of transmission and the delay do not rely on the group scale. The purpose of this scheme is that a data supplier can trace malicious authorized users who gave a decryption key to an unauthorized use. However, this scheme can not be applied to "entity revocation".

Recently, paper [10] has proposed two methods which enable an efficient "entity revocation" of which the amount of transmission and the delay don't rely on the group scale.However, the methods require a preparation phase when a distributor sends each encrypted group key for each entity before deciding a revoked entity. Therefore, the methods are not suitable for a system of which "entity revocation" happens frequently. Moreover, the methods require a fixed and privilege distributor who manages all secret keys of other entities.

In this paper, we propose a group key distribution scheme with "entity revocation" to achieve the following requirements:

-The amount of transmission and the delay, from deciding a revoked entity until completing a group key distribution for all entities, does not rely on the group scale. We believe this requirement is effective to achieve a quick key distribution with "entity revocation" when the group is large.

-Preparation phase when a distributor sends each encrypted group key to the corresponding entity, is not necessary. This requirement is suitable for a system with a frequent "entity revocation".

-The fixed-privileged distributor isn't required. Anyone, called "coordinator" in this paper, can do "entity revocation".

The organization of this paper is as follows. In Section 2, we will explain our approach by examining two methods in paper [10]. In Section 3, we propose our scheme to satisfy the above requirements after explaining our target system. In Section 4, we discuss the security of our proposed scheme. In Section 5, we describe some considerations that are necessary to apply our scheme to an actual system. Also, in Section 6, we evaluate the performance and features of our scheme.

## 2 Approach

In this section, we investigate basic methods seen in paper [10] to fulfill which satisfies a part of our requirements. We modify it into a simpler method because the methods shown in [10] are rather complicated to analyze. And we pick up the above mentioned remaining problems and show our approach to solve them.

### 2.1 Basic Methods from Paper [10]

The methods shown in paper [10] satisfy the first requirement that the amount of transmission and the delay do not rely on the group scale.

The methods consist of two steps. In the first step, a fixed-privileged distributor generates a group key and sends the encrypted group key called "preparation data" here, for each entity, in such a way that any entity has not been able to decrypt it yet. In the second step, the distributor decides which entity should be revoked and broadcasts the secret key of the revoked entity. The amount of transmission and the delay in the second step do not rely on the number of entities. Receiving the broadcast data, all the entities except the revoked one can decrypt the preparation data to get the group key. The methods use a mathematical technique that is known as "RSA common modulus attack" [11] and "RSA low exponent attack"[6] respectively to realize:

-to distinguish a revoked entity from the other entities, and

-to share a same group key among the other entities.

Here we call the method by using technique of [11] "Previous Method 1", the method by using technique of [6] "Previous method 2".

We think their attacks are a type of "secret sharing schemes". However, "RSA low exponent attack" uses Chinese remainder theorem, similar to a secret sharing scheme proposed in paper [1].

Next, we show a modification of the scheme in paper [10], using a general secret sharing technique.

### 2.2 A Modification of The Scheme from Paper [10]

1 We assume there exists a secure communication path between a distributor and each entity, using symmetric cryptography or asymmetric cryptography.

2 The distributor generates a secret data $S$ as the group key, and divides it by threshold 2, using the general secret sharing technique shown in paper [12]. And the distributor sends each shadow $s_i$ to entity $i$ as its secret key, through each secure communication path of step1. So it takes a time relying on the group scale to distribute all the shadows.

3 Let us support the distributor needs to revoke the entity $j$. Then the distributor broadcasts the secret key $s_j$ of entity $j$. Of course, this amount of transmission and the delay don't rely on the group scale.

4 Receiving the secret key $s_j$, all entities except the revoked one can recover the group key $S$ by using two sets of shadow: its own secret key and the secret key $s_j$. The revoked entity alone can't recover the group key $S$ because it can get only one secret key $s_j$.

This scheme can expand so as to revoke $k$-1 entities at a time, by dividing the secret $S$ by threshold $k$.

## 2.3 Our Approach

In both methods seen in paper [10] and in the modified scheme shown above, it is necessary for the distributor to send each preparation data (or shadow) to the corresponding entity. Therefore, it takes a long time to finish distributing the preparation data when the group is large. So we consider that these schemes unsuitable for a system with frequent "entity revocation". We need a scheme that reuses the distributed shadow while maintaining high security high. Also, because the methods in paper [10] use an RSA-like cryptosystem, heavy calculation with long integers is required for each entity. So we need a scheme of which security is based on discrete logarithm problem (DLP), in order to reduce data size and calculation time while maintaining high security, by using elliptic curve cryptosystems (ECC) or hyper- elliptic curve cryptosystems (HCC).

Also, the method in paper [10] and the modified scheme require a fixed-privileged distributor to manage the secret keys of all the entities. We require a scheme whereby any entity can become a distributor in order to apply it to a system where all members have equal rights, like a conference system.

Our approach to achieve the above requirements is as follows:

-We apply "threshold cryptosystems" shown in paper [4] based on DLP to our purpose. This is a type of secret sharing scheme such that the entity can reuse the shadow. According this approach, we expect that the preparation phase is not needed and that the calculation time can be reduced by using ECC or HCC.

-We use the one-pass Diffie-Hellman key exchange scheme to distribute preparation data. According this approach, we expect that any entity can become a distributor.

Now, we explain our scheme under above approaches.

# 3 Proposed Scheme

## 3.1 Target System

The broadcast system of our target is defined as follows:

System manager:

A trusted party who decides system parameters and sets each entity's secret key. Also it manages a public bulletin board.

Entity: $i$

A user or terminal that is a member of the group. We assume the group has $n$ entities, and let $\Phi$ be a set of the entities:

$$\Phi = \{ 1, 2 \dots , n \}.$$

Also, we assume that all entities are connected to a broadcast network and that any entity can send data to any other entities simultaneously.

Coordinator: $v$

A coordinator decides a (or some) revoked entity (ies) and coordinates a group key distribution with "entity revocation". We use the term "coordinator" to distinguish it from the fixed-privileged distributor discussed earlier. In our scheme, any entity can become coordinator.

Revoked entity: $j$

An entity to be revoked by the coordinator. Let $\Lambda$ ($\subset\Phi$) be a set of revoked entities, having $d$ entities.

Public bulletin board:

It keeps system parameters and public keys for all entities with certifications made by the system manager. We assume that any entity can get any data from this board at any time.

Thereafter, we explain our scheme, dividing into system setup phase and key distribution phase.

## 3.2 System Setup Phase

A system manager decides a parameter $k$ that is satisfied:

$$0 \leq d \leq k\text{-}1 < n,$$

where $n$ is the number of entities in the group and $d$ is the number of revoked entities.

1 The system manager decides the following system parameters and publishes them to a public bulletin board:

$p$ : a large prime such that $p > n+k-1$ (about 1024 bit),

$q$ : a large prime such that $q \mid p-1$ (about 160 bit) and

$g$ : a $q$ th root of unity over GF($p$).

The system manager generates a system secret key $S$ ($\in Z_q$), and stores it secretly.

2 The system manager divides the system secret key $S$ into $n+k-1$ shadows by threshold $k$, using well-known Shamir's secret sharing scheme [12]:

1 $a_0 = S$.

2 The system manager defines the following equation over GF ($p$):

$$\mathrm{f}(x) = \sum_{f=0}^{k-1} a_f x^f \bmod q, \tag{1}$$

where $a_1, a_2, \ldots, a_{k-1}$ are random integers which satisfy the following conditions:

$$0 \le a_i \le q-1 \text{ for all } 1 \le i \le k-1 \text{ and } a_{k-1} \ne 0.$$

3 The system manager generates $n+k-1$ shadows as follows:

$$s_i = \mathrm{f}(i) \quad (1 \le i \le n+k-1).$$

3 The system manager distributes the shadows $s_1, \ldots, s_n$ to each entity $1, \ldots, n$ respectively through a secure way. Each entity keeps its own shadow as its secret key. The remaining $k$-1 shadows are safely stored as spare secret keys.

4 The system manager calculates public keys $y_1, \ldots, y_{n+k-1}$ by the following equation:

$$y_i = g^{s_i} \bmod p \quad (1 \le i \le n+k-1). \tag{2}$$

Then the system manager publishes $y_1, \ldots, y_n$ on the public bulletin board with the corresponding entity's identity numbers. The remaining $y_{n+1}, \ldots, y_{n+k-1}$ are published to the public bulletin board as spare public keys.

## 3.3 Key Distribution Phase

<Generation of broadcast data by the coordinator>

First, a coordinator generates a broadcast data B ($\Lambda$, $r$) as follows:

1 The coordinator $v$ calculates the preparation data

$$X = g^r \bmod p, \tag{3}$$

where $r$ is a random number ($\in Z_q$).

2 The coordinator $v$ decides which entities to revoke. Let $\Lambda$ be the set of revoked entities and $d$ is the number of the revoked entities.

3 The coordinator $v$ picks $k$-$d$-1 integers from a set $\{n+1, ..., n+k-1\}$ and let $\Theta$ be the set of chosen integers. Then the coordinator calculates $k$-1 revocation data as follows:

$$M_j = y_j^{\,r} \bmod p \quad (j \in \Lambda \cup \Theta), \tag{4}$$

using the public keys of revoked entities and the spare public keys on the public bullet in board.

4 The coordinator $v$ broadcasts following broadcast data to all entities:

$$\mathrm{B}\,(\Lambda, r) = X \,\|\, \{[\,j, M_j] \mid j \in \Lambda \cup \Theta\}, \tag{5}$$

where || indicates "concatenation" of data.

<Calculation of the group key $U$ by the coordinator>

The coordinator $v$ calculates a group key $U$ using its own secret key $s_v$ and broadcast data B ($\Lambda$, $r$):

$$U = X^{s_v \times \mathrm{L}(\Lambda \cup \Theta \cup \{v\},\, v)} \times \prod_{j \in \Lambda \cup \Theta} M_j^{\mathrm{L}(\Lambda \cup \Theta \cup \{v\},\, j)} \bmod p, \tag{6}$$

where

$$\mathrm{L}\,(\Psi, w) = \prod_{t \in \Psi \setminus \{w\}} t/(t-w) \bmod q \quad (\forall \Psi\text{: set, } \forall w\text{: integer}). \tag{7}$$

Since $M_j$ (= $g^{s_j \times r}$ mod $p$) holds, the system secret key $S$ is recovered on the exponent of equation (6), gathering $k$ sets of secret keys:

$$\begin{aligned} U &= g^{r \times s_v \times \mathrm{L}(\Lambda \cup \Theta \cup \{v\},\, v)} \times \prod_{j \in \Lambda \cup \Theta} g^{s_j \times r \times \mathrm{L}(\Lambda \cup \Theta \cup \{v\},\, j)} \bmod p \\ &= g^{r\{s_v \times \mathrm{L}(\Lambda \cup \Theta \cup \{v\}, v) + \sum_{j \in \Lambda \cup \Theta} (s_j \times \mathrm{L}(\Lambda \cup \Theta \cup \{v\}, j))\}} \bmod p \\ &= g^{\mathrm{r} \times \mathrm{S}} \bmod p. \end{aligned}$$

Each entity can reuse its secret key $s_i$, which is a shadow of the system secret key $S$, because the system secret key $S$ is recovered on exponent of GF ($p$), not on GF ($p$).

<Calculation of the group key $U$ by a non-revoked entity>

Receiving the broadcast data, a non-revoked entity $i$ calculates the group key $U$ using its own secret key $s_i$, similar on the coordinator $v$,

$$\begin{aligned} U &= X^{s_i \times \mathrm{L}(\Lambda \cup \Theta \cup \{i\},\, i)} \times \prod_{j \in \Lambda \cup \Theta} M_j^{\mathrm{L}(\Lambda \cup \Theta \cup \{i\},\, j)} \bmod p \\ &= g^{r \times S} \bmod p. \end{aligned} \tag{8}$$

The system secret key $S$ is recovered on the exponent of equation (8), gathering $k$ secret keys.

On the other hand, a revoked entity *j* can not calculate the group key *U* because $X^{s_j}$ which the entity *j* can calculate by using its own secret key $s_j$ is equal to the revocation data $M_j$, and the entity *j* can gather only *k*-1 secret keys on the exponent.

### 3.4 Concrete Example

We show the concrete example in Figure.1 for the following one:

1 The coordinator (entity 2) decides to revoke the entity 4.

2 The coordinator calculates the preparation data $X$ ($=g^r \bmod p$) and the revocation data $M_4$($=y_4^r \bmod p$).

3 The coordinator broadcasts the broadcast data B (4, *r*) = $X \| \{[4, M_4]\}$.

4 The coordinator calculates the group key $U$ ($=g^{r \times S} \bmod p$) by using its own secret key $s_2$ and the broadcast data B (4, *r*).

5 The Entity 1 calculates the group key *U* by using its own secret key $s_1$ and the broadcast data B (4, *r*).

6 The Entity 3 calculates the group key *U* by using its own secret key $s_3$ and the broadcast data B (4, *r*).

7 The Entity 4 can't calculate the group key *U* by using its own secret key $s_4$ and the broadcast data B (4, *r*). Because the broadcast data B (4, *r*) includes the secret key $s_4$.

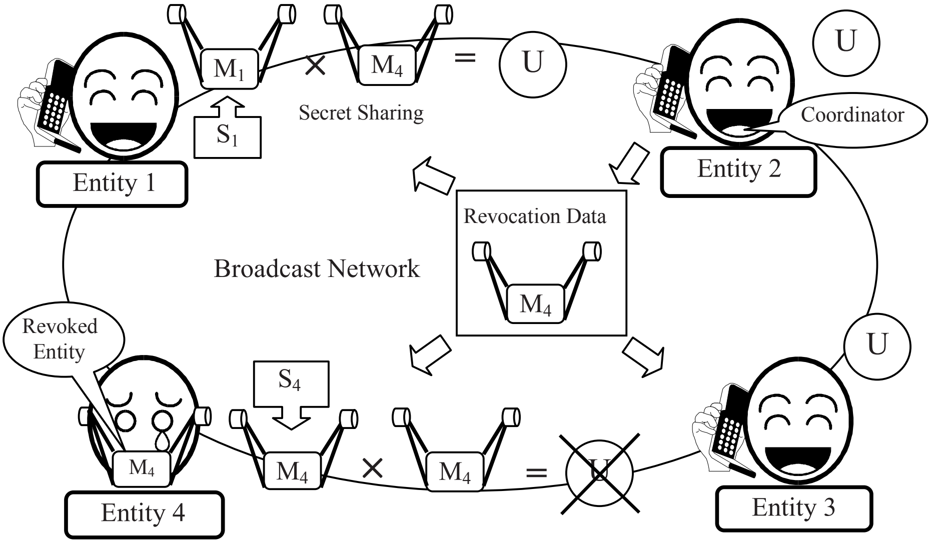

Fig.1. The concrete example of our proposal

## 4 Security

First, we discuss how difficult is finding the group key $U$ for the revoked entity and an outsider of the group. We will examine three types of attacks to pillage the group key as follows:

1 All entities can get $y$ (=$g^S$ mod $p$) by using $k$ sets of public key $y_i$ shown in equation (2). To get the group key $U$ from $y$, the revoked entity needs to obtain the random number $r$ that the coordinator generates. Since the random number $r$ is an exponent of the preparation data and the revocation data, the level of difficulty in getting $r$ is the same as that of solving DLP.

2 To get the group key $U$ from the preparation data $X$, the revoked entity needs to obtain the system secret key $S$ that the system manager generates. Since the system secret key $S$ is an exponent of the above $y$ (=$g^S$ mod $p$), the level of difficulty of getting $S$ is also the same as that of solving DLP.

3 We assume a trial to get the group key $U$ from the revocation data shown in equation (4). From broadcast data B ($\Lambda$, $r$), all entities can get $k$-1 revocation data $M_j$ which includes the secret key $s_j$ respectively on the exponent. Since the system secret key $S$ is divided into the secret keys by threshold $k$, however, the revoked entity can not calculate the group key $U$ which includes $S$ on the exponent. Even if the revoked entity uses its own secret key $s_j$, the number of shadows does not increase.

Next, we consider an attack to modify and forge broadcast data. The coordinator generates the preparation data $X$ and the revocation data $M_j$ using only the public information. Therefore, it is necessary to append the coordinator's signature to the broadcast data in order to prevent modification and forgery. In sections 5 and 6, we will explain a method that includes coordinator authentication, in which the amount of broadcast data does not increase, compared with the basic scheme explained in section 3. Moreover, we consider that a time-stamp on the broadcast data is necessary to prevent a replay attack. To prevent an attacker from modifying and forging a public key on a public bulletin board, the board should be managed by a trusted system manager or all public information should be stored with certifications made by a trusted third party.

Finally, we discuss the security of our scheme when entities form a conspiracy. Even if all revoked entities conspire, they can not reconstruct the secret key $S$ since they can get at most $d$ ($< k$ ) shadows $s_j$ of $S$, which is less than the threshold-$k$. Here, we don't assume a conspiracy attack that a non-revoked entity cooperates with the revoked entity. If the attack is possible, revoked entities can get all group keys and all decrypted messages through the co-conspirator. To prevent this type of attack, different techniques are required, for example traitor tracing or watermark, which is outside of the scope of this paper.

# 5 Applications

In this section, we describe some considerations that are necessary to apply our scheme to an actual group communication system.

## 5.1 New Entity

When a new entity wants to join a group communication system, a system manager decides its unique identity number $c$ ($n$+$k \leq c \leq q$-1) which is different from ones of the existing entities. The system manager calculates its secret key $s_C$= f ( $c$ ) and sends it to the new entity through a secure way. Then, the system manager calculates the public key $y_C$ (= $g^{s_c}$ mod $p$) and adds it to the public bulletin board.

This procedure does not affect the existing entities.

## 5.2 The Number of Revoked Entities

Our method shown in section 3 enables a coordinator to revoke a maximum of $k$-1 entities at one time. Also, the parameter $k$ determines the amount of broadcast data from equation (5). If the number of entities that the coordinator can revoke at once becomes large, the broadcast data amount becomes large. Therefore, a system manager should decide the parameter $k$ to fit for an actual system. The coordinator can distribute a group key without "entity revocation", by using $k$-1 sets of spare public information for the broadcast data.

## 5.3 Continuity of Revocation

In actual group communication, our scheme is used repeatedly by a different coordinator and revoked entities. A coordinator can decide either one of the following cases:

-revoke entities that were revoked last time (by indicating the entities as revoked again) or

-send a new group key to entities which were revoked last time (by not indicating the entities as revoked this time).

Also, the coordinator can use the previous group key to make a new one in order to revoke entities that were revoked last time.

Next, we show a method of revoking specific entities from the group communication completely:

1 The system manager distributes a random number $e$ to all entities other than the specific ones by our scheme shown in section 3.

2 Each entity except the specific ones replaces own secret key $s_i$ with

$$s_i' = s_i \times e \mod q. \tag{9}$$

3 The system manager replaces the system parameter $g$ on the pubic bulletin board with

$$g' = g^{1/e} \bmod p. \tag{10}$$

With this method, it is not necessary to change every public key on the public bulletin board since $y_i = (g')^{s_i'} \bmod p$ is satisfied. It is practical because the public keys might be stored in a local storage by each entity. The revoked entities do not join the group communication permanently because they do not have the secret key $s_j'$ to satisfy $y_j = (g')^{s_j'} \bmod p$. When the system manager wants the revoked entity to join the group communication again, the system manager would send its new secret key $s_j'$ through a secure way.

## 5.4 Some Modifications

Our scheme is considered a one-pass Diffie-Hellman key exchange scheme with "entity revocation". The coordinator distributes the preparation data $X$ $(=g^r \bmod p)$, and shares a group key $U$ $(=g^{S \times r} \bmod p)$ with the other entities, where we regard $y$ $(=g^S \bmod p)$ as a public key for the group. Thus, a coordinator can select any group key $Z$ by broadcasting $V$ $(=Z \times U \bmod p)$ together, similar to the ElGamal public key cryptosystem [5].

If an attacker can use a key calculation mechanism of an entity as an oracle, a similar modification using the Cramer-Shoup public key cryptosystem [3] would be effective against an adaptive chosen ciphertext attack.

Also, we can modify our scheme so as to prevent an attacker from modifying and forging broadcast data, by adding a message recovery signature as follows:

<Setup by a system manager>

Same as the basic scheme explained in section 3 except that the system manager publishes a hash function (hash) on a public bulletin board.

<Generation of broadcast data by a coordinator>

1 The coordinator calculates revocation data $M_j$ for $j \in \Lambda \cup \Theta$ shown in equation (4), by using a random number $r$.

2 The coordinator calculates a following hash data:

$$H = \text{hash}\,(v \,\|\, [\, j \,\|\, M_j\,],\ j \in \Lambda \cup \Theta). \tag{11}$$

3 The coordinator $v$ generates its signature of the hash data, using its secret key $s_v$:

$$Z = H \times (-s_v) + r \bmod q. \tag{12}$$

4 The coordinator $v$ broadcasts the following broadcast data:

$$\mathrm{B}\,(\Lambda, r) = Z \,\|\, v \,\|\, \{\, [\, j, M_j] \mid j \in \Lambda \cup \Theta\}. \tag{13}$$

The amount of broadcast data is less than that of our basic scheme explained in section 3 shown in equation (5) since $Z \,\|\, v$ is less than $X$ (1024bit).

<Key exchanging by a non-revoked entity>

1 Similar to equation (11), the entity calculates hash data $H'$. If the data is not changed, $H' = H$.

2 The entity recovers the preparation data $X'$ using the public key of the coordinator $y_v$:

$$X' = g^{Z} \times y_v^{H'} \mod p. \qquad (14)$$

If the signature $Z$ originates from the right coordinator $v$, $X' \equiv X$ shown in equation (3).

The rest of procedure to distribute the group key $U$ is the same as the basic scheme explained in section 3.

This scheme uses one of the message recovery signature schemes proposed in [13]. Therefore, other variations of the signature are possible:

For example, $Z' = H \times r + s_v \mod q$.

## 6 Evaluation

In this section, we will evaluate our proposal, comparing with the following four previously reported methods:

-Familiar Method 1: a method whereby a distributor distributes a group key to $n$-$d$ entities individually, except $d$ revoked entities, using a 128bit symmetric key block cipher.

-Familiar Method 2: the same method as Familiar Method 1, using 1024bit RSA cryptosystems.

-Previous Method 1: a method in paper [10], using the "RSA common modulus attack".

-Previous Method 2: another method in paper [10], using the "RSA low exponent attack".

We will evaluate our proposal based on the four requirements that we have already described in sections 1 and 2:

-Requirement 1: The amount of transmission and the delay do not rely on group scale.

-Requirement 2: Preparation phase is not necessary.

-Requirement 3: The fixed-privileged distributor is not required.

-Requirement 4: The security of the scheme is based on DLP.

First, we will evaluate the performance of our proposal. Figure 2 shows the performance of our proposal, compared with "Familiar Method 1" and " Familiar method 2". In Figure 2, axis x shows the number of entities in the group, and axis y marks the delay (sec) until all entities complete a group key sharing. However, the number of revoked entities is $d$=1. We assume that the delay is the sum of data transfer time and calculation time for each entity. We estimate data transfer time by assuming the transmission rate to be 28.8kbps. Also, we estimate a calculation time by using experimental results obtained from a 200MHz Sun Ultrasparc (with gcc 2.7.2.3).

This figure shows the delay of our proposal does not rely on group scale. So, our scheme satisfies requirement 1 shown above. In Figure 2, "Familiar Method 1" seems more efficient than our proposal, because the cross-point of them is rather large (*n*=180). Though we estimate the data transfer time by the amount of transmission here, the data transfer time is related to the number of communication in actual communication. Some control data is added to the transmitted data for each connection. Also an authentication and a negotiation are necessary for each communication. Therefore, we consider the cross-point of two methods is surely much less than *n*=180, because the number of communication of "Familiar method" increases, relying on the group scale. On the other hand, the number of communication of our proposal is constant.

Figure 3 shows the performance of our proposal, compared with "Previous Method 1" and "Previous method 2". In Figure 3, axis x shows the number of revoked entities *d*, and axis y marks the delay (sec) until all entities complete a group key sharing. Here, measurement conditions are the same as in Figure 2. "Previous Method 1" can not revoke two or more entities at once.

We can see that the delay of our proposal is less than that of "Previous Method 2" where *d*≥45. Therefore, our proposal can revoke entities quickly, even if a coordinator should revoke many entities at one time. Moreover the calculation time of "Previous Method 2" increases exponentially as *d* increases*,* On the other hand, the operation amount of our proposal increases linearly as *d* increases. The delay of our proposal is within 1 sec in the case of *d*=1.

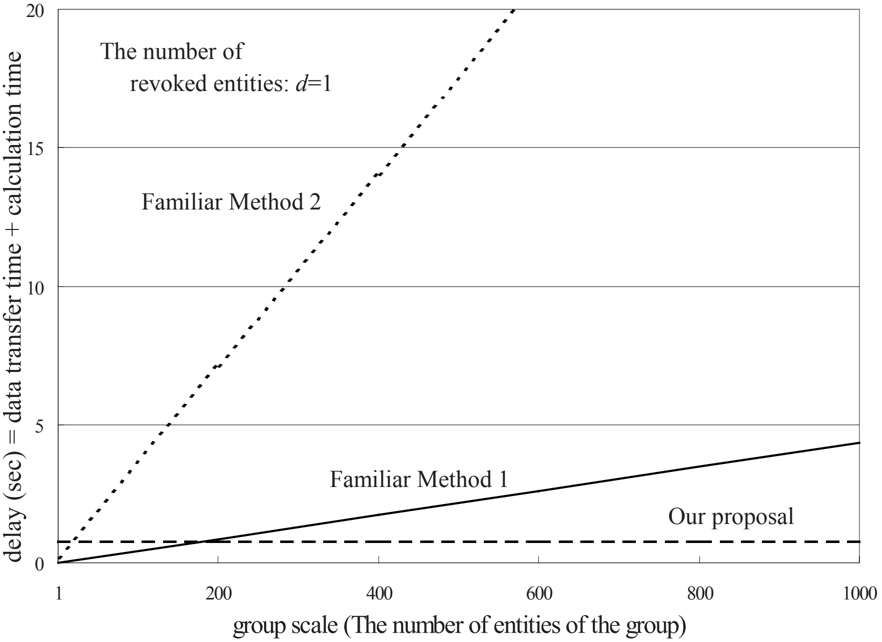

Fig.2. Performance Comparison (with Familiar Methods)

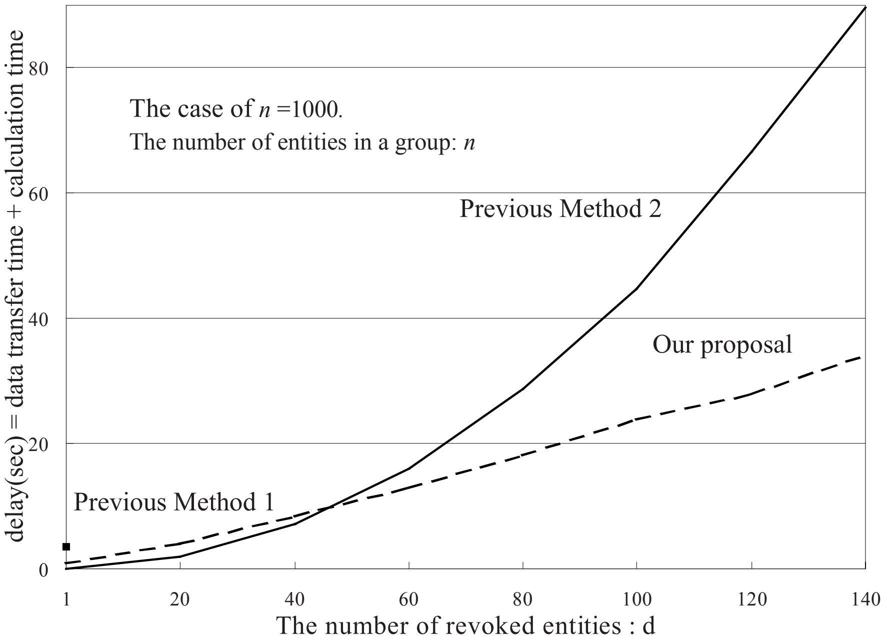

Fig.3. Performance Comparison (with Previous Methods)

Next, we evaluate the features of our proposal. Table 1 shows that our proposal satisfies four requirements, whereas four existing methods do not. Therefore, we consider that our proposal can be applied to many systems with some restrictions.

Table1. Comparison of features

| Requirement<br>Method | Requirement 1 | Requirement 2 | Requirement 3 | Requirement 4 |
|---|---|---|---|---|
| Our proposal | **yes** | **yes** | **yes** | **yes** |
| Familiar Method 1 | no | **yes** | no | |
| Familiar Method 2 | no | **yes** | **yes** | no |
| Previous Method 1 | **yes** | no | no | no |
| Previous Method 2 | **yes** | no | no | no |

## 7 Conclusions

In this paper we have proposed a quick group key distribution scheme with "entity revocation". The features of our scheme are as follows:

-The amount of transmission and the delay do not relay on group scale. This feature allows a quick key distribution with "entity revocation" even when the group is large.

-Preparation phase is not necessary. This feature is suitable for a system with frequent "entity revocation".

-Any entity can act as coordinator, and revoke any other entities. This feature is suitable for group communication systems in which all members have equal rights like a conference system.

-Data transfer time and entity calculation time are reduced by using elliptic curve or hyper-elliptic curve cryptosystems.

## References

[1] C.Asmuth, J.Bloom, "A Modular Approach to Key Safeguarding", IEEE Trans. on Information Theory, v. IT-29, n.2, Mar 1983, pp.208-210.

[2] M.Burmester, Yvo.Desmedt, "A Secure and Efficient Conference Key Distribution System", Advances in Cryptology-EUROCRYPT'94, pp.275-285, Springer-Verlag, 1994.

[3] R.Cramer, V.Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack", Advances in Cryptology: Proceedings of CRYPTO'98, pp. 13-25, Springer-Verlag, 1986.

[4] Y.Desmedt, Y.Frankel, "Threshold cryptosystems", Proceedings of Crypto'89, LNCS435, pp.307-315, Springer-Verlag, Aug. 1989.

[5] T.ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. Inform. Theory, 31: 469-472, 1985.

[6] J.Hastd, "On using RSA with low exponent in a public key network", Advances in Cryptology: Proceedings of CRYPTO'85, Vol.218, pp. 403-408, Springer-Verlag, 1986.

[7] M.Hwang, W.Yang, "Conference key distribution schemes for secure digital mobile communications", IEEE Journal on Selected Areas in Communications, vol. 13, No.2, Feb. 1995.

[8] I.Ingemarsson, D.T.Tang, and C.K.Wong, "A conference key distribution system", IEEE Trans. Inform.Theory, vol. IT-28, pp. 714-p. 720, Sep. 1982.

[9] K.Kurosawa, Y.Desmedt, "Optimum Traitor Tracing and Asymmetric Scheme", Advances in Cryptology-EUROCRYPT'98, pp. 145-157, Springer-Verlag, 1998.

[10] N.Matsuzaki, J.Anzai, "Secure Group Key Distribution Schemes with Terminal Revocation", Proceedings of 1998 First Japan- Singapore Joint Workshop on Information Security, pp. 37-44, 1998.

[11] G.J.Simmons, "A 'Weak' privacy protocol using the RSA cryptosystem", Crypto-logia, Vol.7, No.2, pp. 180-182, 1983.

[12] A.Shamir, "How to share a secret", Comm.Assoc. Comput. Mach., vol. 22, no. 11, pp. 612-3, Nov. 1979.

[13] K.Nyberg, R.A.Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem", Proceedings of Eurocrypt'94, LNCS950, pp. 182-193, 1995.