The ID-based Non-interactive Group Communication Key Sharing Scheme using Smart Cards

Hiroyuki Sakakibara  Kazunori Seki  Ken-ichi Okada  Yutaka Matsushita

Department of Instrumentation Engineering, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223, Japan
Email : hiroyuki@myo.inst.keio.ac.jp

Abstract

Attention to CSCW has been increased recently, and it derived the needs of the secure group communication. In order to realize secure group communication, data which is sent from a member to other members of a group should be encrypted by the cryptographic communication key of the group. In this paper, we propose an identity-based non-interactive group communication key sharing scheme using smart cards based on the MCK method. We assume that smart cards which contain key generators and are secure for tampering. Each user has a smart card and a key generator. A user can generate a group communication key non-interactively with his key generator and IDs of other group members of the group using his smart card.

1 Introduction

Recently, attention to computer supported cooperative work has increased[1][2]. In the group communication which the member of a group in cooperative work communicates with other members of his group, it will be desirable that members of the same group can communicate securely with each other. In an organization, there may be a member who belongs to several groups, because we usually have several works at the same time. In the case of managing groups we will have to consider the change of group structure, since new working groups may be organized frequently and some groups may be re-organized with some flexibility. Practically, such changes of group structure mentioned above can be seen often in our organizations.

In order to realize secure group communication, important data which is sent from a member of a group to other members of the group should be encrypted by a common cryptographic key of the group(i.e., group communication key).

Non-members of the group can not understand the data which is encrypted by the group communication key of the group. Considering group-oriented environment mentioned above, cryptographic key sharing scheme which is suitable for group communication should be proposed.

The Copy Key(CK) method[3] is a typical cryptographic key management method for group communication, where keys are assigned to the members of groups. But this method has serious problems in terms of the key renewing and re-distribution. In order to change the structure of groups in an organization the key must be renewed and re-distributed to the members of the changed (or new) group, thus we cannot say that it is suitable for group-oriented structure. We have proposed the Modified Copy Key(MCK) method to conquer this problem[4][5][6]. But this method is not secure for a conspiracy attack.

On the other hand, an interactive key sharing scheme between two entities has been proposed in [7] and several Identity-based Non-Interactive Key Sharing schemes in which any pair of entities(e.g., users) can share a same cryptographic key non-interactively(IDNIKS) have been proposed in [8][9][10]. The IDNIKS has two properties which are more excellent than [7] in key sharing, namely, 1st. non-interactive, 2nd. using IDs of users and this is very attractive feature. However in these IDNIKS schemes, more than three users can not share a same key non-interactively. Identity-based conference key distribution schemes which are suitable for group communication has been proposed in [11], however this is not complete non-interactive key sharing scheme for group communication. Security of a shared key is studied well in these schemes, however we can not say that these schemes are complete non-interactive key sharing schemes which are suitable for group-oriented
environment.

In this paper we propose an identity-based non-interactive key sharing scheme which is suitable for group-oriented environment using smart cards. Our scheme which is based on the MCK method and the concept of the IDNKS, also excellent in managing of keys. In Section 2, a summary of the MCK method is described. And in Section 3, our key sharing scheme for group-oriented environment is described. In Section 4, we discuss our scheme and conclusion is given in Section 5.

2 The Modified Copy Key(MCK) method

The Modified Copy Key(MCK) method is a group-oriented key management method, we have proposed in [4][5][6]. It overcomes the demerit of the Copy Key(CK) method[3] in key management and it is very flexible in generating various group communication keys. Our new scheme in this paper is based on the MCK method partly.

In this section, we describe summaries of the CK method and the MCK method.

2.1 The Copy Key method

The Copy Key(CK) method is a typical conventional cryptographic key management method for group communication. Keys are assigned to the members of each group beforehand and the members make cryptographic communication with these keys. However, this method has serious problem in terms of key renewal and re-distribution.

![Diagram](image)

**Figure 1: The Copy Key method**

In order to change the structure of groups, renewed group communication keys must be re-distributed to members of the renewed groups. For example, in Fig 1, assuming that the members of C, M2 and M3 are removed from group C, a renewed key must be re-distributed to the rest of members of C, M4 and M5. Since these changes of members in groups seem to occur very often in an organization, renewal and re-distribution of keys are some troubles for a key manager of the organization.

2.2 The Modified Copy Key(MCK) method

The Modified Copy Key(MCK) method is a cryptographic key management for flexible group communication, that is, in this scheme a user can share a same group communication key with more than 2 users non-interactively. And it is much better in management of keys than the CK method, but it is vulnerable to a conspiracy attack.

2.2.1 The basic idea of MCK

We assume a communication system where many groups are formed and each user belongs to one or more groups. When let n be the number of all users, there are specified n Pieces P1, P2, ..., Pn which are informations for generation of group communication keys (i.e., key generators). Each user holds a common key and a unique set of Pieces obtained by combinations of n Pieces taken n - 1 at a time, then a member Mj and another member Mj share n - 2 unique Pieces. Note that a Piece is a positive integer, and Pi ≠ Pj if i ≠ j. A distribution of the common key and Pieces for a group is illustrated in Fig 2, where n equals to 6. These Pieces are distributed to users by a trusted key manager.

![Diagram](image)

In Fig 2 the members of a group M1, M2, M3 have P3, P5 and P6, on the other hand, the non-members M4, M5, M6 do not have all of P3, P5 and P6. Since M1, M2 and M3 can generate a key from the common key K0 and Pieces P3, P5, P6, they can share a secure cryptographic communication key with each other non-interactively. When a user wants to broadcast to all users, they use a common key K0 for it. Consequently each member can communicate with other members of his group(s) by managing one common key and n - 1 Pieces.

2.2.2 Key generation

In the MCK method, users don't hold group communication keys used for enciphering and deciphering, but hold a common key and key generators "Pieces" which are positive integers instead of keys themselves. Each user gets a communication key generated from a common key and Pieces held by himself, when he makes cryptographic communication with other users.
<table>
<thead>
<tr>
<th>P₁</th>
<th>P₂</th>
<th>P₃</th>
<th>P₄</th>
<th>P₅</th>
<th>P₆</th>
<th>K₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>∗ M₁</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>∗ M₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>∗ M₄</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M₅</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M₆</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**M₁** Member i  
**P₁** Piece i  
**K₀** Common Key  
● A common piece for a group communication  
○ A distributed piece  
∗ A member of the group  
{M₁, M₂, M₄}

**Figure 2:** distribution of Pieces

A plain text is encrypted and decrypted by the group communication key. The group communication key $K$ can be generated from a common key $K₀$ and a Piece $P$ as follows:

$$K = f_{MCK}(K₀, P)$$  

(1)

where $f_{MCK}()$ is an appropriate encryption function which generate a unique group communication key from $P$ and $K₀$. When multiple $P\{P₁, P₂, ..., Pₖ\}$ are used to generate a group communication key $K$, it is generated as follows:

$$K₁₂₃...i = f_{MCK}(..f_{MCK}(f_{MCK}(K₀, P₁), P₂), ..., Pᵢ)$$  

(2)

where $P$ is substituted in the ascending order of the subscript number of $P$. $f_{MCK}()$ is one to one mapping, therefore a unique group communication key is generated from a unique set of $Ps$. Let $n$ be the number of users in a group, group communication keys are generated from the combinations of $n - 1$ Pieces taken $r(2 \leq r \leq n - 1)$ at a time. Consequently the total number of keys which can be generated from $n - 1$ Pieces is as follows:

$$\sum_{r=1}^{n-1} n - 1Cr = 2^{n-1} - 1$$  

(3)

Hence a user can manage $2^{n−1}$ keys (one common key and $2^{n−1}−1$ communication keys) by holding one common key and $n−1$ Pieces.

This paper omits encryption and decryption procedure (refers to [6]).

### 2.2.3 The problem of the MCK method

If a legal user knows his own Pieces and their subscript numbers, it will be possible to get other Pieces from another key in which his own Pieces are involved. For example, if the legal user who holds $P₁$ and $P₂$ gets $K₁₂₃$, he might be able to get $P₃$ by means of the known-plain-text attack. Hence the transmission is kept safe from a legal user's attack. Note that legal users indicate users who are assigned Pieces legally by a trusted key manager in an organization.

We should pay more attention to a conspiracy attack by legal users. If more than two users conspire with each other, they would be able to get the common key $K₀$ and all Pieces easily. As a result they can generate all group communication keys from them. Therefore the cryptosystem in which any user can not handle common key and Pieces directly is required. This condition will be satisfied if the encryption system and decryption system are realized as a function box in which the detail system configuration is hidden from users.

### 3 A non-interactive IDentity-based group communication key sharing scheme based on the MCK method using smart cards

In the MCK method, if more than two users(members) show their Pieces with each other, they can know all Pieces and can make any group communication key. However, in the MCK method, users can share a same key among more than two users non-interactively, once Pieces are distributed to them. This is a remarkable feature when users(members) communicate in various groups securely.

In order to defend against conspiracy attack which is described in Section 2.2.3, Pieces which are distributed and held by users must not be exposed to them. In order to realize this requirement, we assume that Pieces of the MCK method are hidden in smart cards, and they are handed to users by the trusted center(TC) in an organization. Note that the TC is an entity that is trusted by all users in an organization and manages all of security informations for all users.

Renewal of Pieces is very important to keep group communication keys secure, because Pieces are key generators. However, in order to renew Pieces in smart cards, the TC has to collect all smart cards from users at the same time. Thus, if the number of users in an organization is large, the trusted center TC cannot renew Pieces in smart cards of users so frequently.

Considering this problem in renewal of Pieces, in
this section, we propose a new group communication key sharing scheme which is based on the MCK method using smart cards.

3.1 Informations for our group key sharing scheme

We define parameters and informations first, then, describe our key sharing scheme as follows.

\[
\begin{align*}
\text{definition} & \\
U_i &  \text{user } i \\
ID_i &  \text{IDentity of } U_i \text{ (a positive integer)} \\
TC &  \text{the Trusted Center in an organization} \\
Nc &  \text{secret information of } TC \text{ (a large positive prime number)} \\
X &  \text{secret information of } TC \text{ (a primitive element over } GF(Nc)) \\
SI_i &  \text{Secret Information of } U_i \text{ (a positive integer, } SI_i = X^{ID_i \mod Nc} \neq 0) \\
MCKS &  \text{the MCK System which works on smart cards and realize the MCK method.} \\
\text{(The MCKS includes Pieces and } f_{MCK}(\cdot)) & 
\end{align*}
\]

IDs are made public to all users in an organization. The TC is in an organization and manages key generators. The TC makes SI_i and send it to user U_i through computer network securely (e.g., using asymmetric cipher system such like RSA[12]). Also the TC hands a smart card to user U_i. A smart card hides the MCKS, functions f_1, f_2, f_3, inside. These are shown in Fig 3 (these functions are explained after).

We assume that MCKS works as follows. For example, let G be a group which is composed of users U_i, U_j, and U_k. When U_i generates a group communication key of G, the MCKS of his smart cards works as:

1. Let IDs be a sequence of other group members, namely, ID_j, ID_k. U_i inputs IDs into his smart card.

2. Pieces and a function \( f_{MCKS}() \) are in the MCKS. The MCKS chooses proper Pieces for the group G which is composed of U_i, U_j, U_k. Then it generates \( K_{ij,k} = G_{MCKS} \) by following (1)(2) in Section 2.2.2, using a function \( f_{MCKS} \). \( K_0 \) and choiced Pieces. These are following the MCK method.

3. The MCKS outputs \( G_{MCKS} \).

We assume that the \( f_{MCK}(\cdot) \) is an one-to-one mapping function and is not time-consuming.

Note that let \( n \) be the number of members(users) in an organization, then the distribution of Pieces for U_i is described in the following way:

\[
P_j : P_{1}, P_{2}, ..., P_{i-1}, P_{i+1}, ..., P_n \quad [P_j \text{ is a positive integer where } j \neq i, 1 \leq j \leq n] 
\]

3.2 Key sharing

In Fig 3, as an example, we assume that a group G in an organization is composed of three users U_i, U_j, and U_k, and they share a same cryptographic group communication key.

Users of a group G share a same group communication key in the following way.

\[U_i's \ procedure] \]

1. U_i inputs SI_i, ID_j and ID_k(IDs) into his smart card. And he inputs m which is a necessary length of the group communication key.

2. The MCKS decides necessary Pieces for the group key of the group G from these IDs.

3. The MCKS generates \( G_{MCKS} \), which is a common information for the group G(subscript G indicates group G). Let P_1, P_2, ..., P_m be the choiced Pieces for G, and \( f_{MCK}(\cdot) \) be the same function which is described in Section 2.2, \( G_{MCKS} \) is described as follows:

\[
f_{MCK}(\cdot) = f_{MCK}(f_{MCK}(K_0, P_1), P_2), ..., P_m) \]

These procedures are following the MCK method in Section 2.2.

4. The function f_1 calculates

\[
f_1(SI_i, IDs) = SI_i^{ID_i \mod Nc} = X^{ID_i \mod Nc} = GI_G \]

(a Group Information for group G and subscript G indicates group G).

Nc is hidden in the f1 as a parameter.

5. The function f_2 calculates

\[
f_2(GI_G, G_{MCKS}) = GI_G \mod G_{MCKS} = G_{K_G} \neq 0 \]

(a Group communication Key for G, the subscript G indicates group G)

6. A function f_3 is a hash function which reduces \( G_{K_G} \) into m bits. That is,

\[
f_3(G_{K_G}, m) = G'_{K_G} \]

7. U_i's smart card outputs \( G'_{K_G} \).

We assume that Nc is hidden in each card securely(actual hidden in the f1 as a parameter of it). Note that Nc is larger than \( G_{MCKS} \).
[Uj’s procedure]
Uj follows Uj’s procedure as well. Note that he must replace the index “i” of “j”.

[Uk’s procedure]
Uk follows Uj’s procedure as well. Note that he must replace the index “i” of “k”.

Following above procedures, users Uj, Uj, Uk can share a same key non-interactively. Namely, the key is described as follows:

\[ G_{K_G} = G_{I_G} \mod G_{MCK_G} \]
\[ = (X^{ID_j}ID_jID_s \mod Nc) \mod G_{MCK_G} \]
\[ \quad \quad \quad \quad \quad \quad \quad \text{[Nc is larger than } G_{K_G}] \]

(9)

\[ G_{K_G'} = f_3(G_{K_G}, m) \]  \hspace{1cm} (10)

Only Uj, Uj, Uk can generate
\[ GI_G = X^{ID_j}ID_jID_s \mod Nc \] from their SIs and IDs.
And only they can generate \( G_{MCK_G} \) by following the MCK method.

Therefore, any Uj where \( l \neq i, j, k \) can not generate \( G_{MCK_G} \) and \( GI_G \), thus he can not generate \( G_{K_G}(G_{K_G'}) \). Once a user generates a group communication key, he can apply it to any symmetric cryptosystem (e.g., DES[13], FEAL[14]).

4 Discussion
4.1 Security

\( G_K \) is calculated from \( GI \) and \( G_{MCK_G} \). \( GI \) is generated from \( SI \) and \( ID \)s where \( SI \) involves \( X \) and \( Nc \). \( G_{MCK_G} \) is generated by the MCKS in smart cards of users.

Hence, in order to discuss security of our scheme, we should discuss two parts, analysis of \( X \) and \( Nc \) and analysis of the MCKS.

4.1.1 Analysis of \( X \) and \( Nc \)
In our scheme, if a user knows \( X \) and \( Nc \), he can make any \( GI \) easily. Hence we discuss the analysis of \( X \) and \( Nc \) by users in an organization.

[Against single user attack]
We assume that the group \( G \) is composed of three users \( U_i, U_j \) and \( U_k \). And \( U_m (m \neq i, j, k) \) who is a legal and not a member of the group \( G \) tries to make \( G_{K_G'} \). In order to make \( G_{K_G'} \), he must generate \( GI_G \) and \( G_{MCK_G} \). If size of \( X \) and \( Nc \) are large enough, it is quite difficult for \( U_m \) to analyze \( X \) and \( Nc \) by himself. Namely, he must find a pair of \( X \) and \( Nc \) which satisfy \( SI_m = X^{ID_m} \mod Nc \). Because he knows only \( SI_m \) and \( ID_m \), he needs an exhaustive search to find out both \( X \) and \( Nc \) from only them.

[Against conspiracy attack]
If two users \( U_m \) and \( U_n \) conspire and show \( SI_m \) and \( SI_n \) each other, can they found out \( X \) and \( Nc \) ?. If they know \( Nc \), they can calculate \( X \) easily by the Euclid attack[15]. However \( Nc \) is hidden in the function \( f_1 \) as a parameter, \( X \) can not be exposed easily. Even if they can know \( Nc \) and \( X \), as a result, can make any \( SI \), they must know all pieces in their MCKS in order to make \( G_{MCK_G} \).

Thus, they have to analyze the MCKS in order to generate \( G_{MCK_G} \) which is the generator of \( G_{K_G} \).
4.1.2 Analysis of the MCKS
Security discussion of the MCK method is mentioned in [6]. Note that we assume that Pieces are hidden in the MCKS and users can not know their Pieces.

If a user or users attack a group communication key of other group, they have to know all of $X$, $N_c$, and all Pieces. Therefore smart cards must have structure which is secure for tampering by users.

In order to keep a group communication key $GK'(or GK')$ secure, $G1$ and $GK_{MCK}$ must be kept secure. In order to keep $G1$ secure, the trusted center $TC$ should renew $N_c$ which is hidden in the function $f_1$ as a parameter in smart cards of all users, and $SI$ which involves a secret information of the TC “$X$”. Also the $TC$ should renew the Pieces in smart cards of all users in order to keep $GK_{MCK}$ secure.

However the $TC$ cannot renew contents of smart cards, namely Pieces and $N_c$ so frequently, because $TC$ have to collect all cards from users at the same time. Therefore, the $TC$ should renew $X$ and make new $SI$, then send it to each user through computer network frequently. Frequent renewal of $SI$(i.e., renewal of $X$) makes $G1$ secure, as a result, it makes $GK$ secure.

When $TC$ renews $N_c$ and Pieces in smart cards, it had better renew a function $f_{MCK}()$ and a common key of the MCK method $K_0$ to keep group communication keys generation much secure.

4.2 Management of Pieces and SI

If an organization is composed of n users, a user in the organization manages n-1 Pieces, one common key of the MCK method $K_0$ and one $SI$. Thus the number of key generators which a user must hold is $n+1$. The n-1 Pieces, $K_0$, are hidden in his smart card. The MCKS in each smart card is secure for tampering, consequently length of Pieces can be small.

4.3 Renewal of key generators

As mentioned above, in order to break security of this scheme, attackers must succeed in finding out $X$, $N_c$, and Pieces($K_0$). Thus, renewal of key generators($X$, $N_c$, Pieces) is very important to keep security of group communication keys. Since the $TC$ can not renew Pieces and $N_c$ so frequently as mentioned above, The $TC$ should renew $X$ and generates a new $SI$, then, sends it to each user through a computer network (note that renewal of $X$ is equal to renewal of $SI$). Let $Ts$ be a renewal period of Pieces and $N_c$ which are in smart cards and let $Tx$ be a renewal period of $X$. Then the $TC$ can set $Ts$ longer than $Tx$, for example, $Tx = 7$days and $Ts = 1$ year.

4.4 Calculation

In our scheme, a smart card must calculate $S1_{1D_1...1D_n} \mod N_c$, namely a modular exponentiation operation. This is the most time-consuming routine for our scheme. Some papers discuss effective modular multiplication method based on software for smart cards[16][17][18]. These papers are effective to realize a typical asymmetric cryptosystems, for instance, RSA[12], which has to calculate large length of modular multiplication repeatedly(note that modular exponentiation is realized by executing modular multiplication repeatedly), for example, 512bits $\times$ 512bits $\mod$ 512bits. Currently, when typical smart cards realize RSA cryptosystem, it takes several seconds to encrypt(decrypt) a 512bits long plain-text[19]. Thus, the $TC$ should choose effective length of $X$ and $N_c$ for modular multiplication(exponentiation), considering security and the execution time.

Considering above state, in order to shorten the execution(calculating) time of this routine, the $TC$ set length of $X$, $N_c$, $ID$ shorter(than 512bits long). An ID can be shorten by applying a proper function which reduce the length of it into shorter.

However, if the $TC$ set these parameters shorter, it will cause another problem. Namely, it is expected that a user who has his $SI$ can find out a pair of $X$ and $N_c$ which satisfies his $SI$ faster. It can be said that relationship between security and execution time is trade-off. The next generation, smart cards will have sufficient computation power and memory capacity of various applications and the appropriate security services such as RSA, and we assume this condition, in order to realize our scheme.

So we propose two practical method which shorten the execution time of the modular exponentiation part in our scheme, considering current power of smart cards.

A:Using Confounders
We describe a method which shorten length of $X$ and $N_c$ securely using a very large random number. Note that functions in smart cards of users is the same as in Fig 3 and Section 3.2

1. The $TC$ generates same informations of Section 3.1 (i.e., $N_c, X, SIs, Pieces(K_0)$). In addition to them, it generates a large random number “$Conf_1(Conf_0)$” for a user $Ui$, whose length is much longer than $X$ and $N_c$ (i=1,...,n, where n is the number of users in an organization). (For example, $X$ and $N_c$ are 100bits long and $Conf_i$ is 500bits long.) And the $TC$ hides it in his smart cards with other key generators(i.e.,
Pieces($K_0$) of the MCKS) secretly. Note that $Conf_i \neq Conf_j$, if $i \neq j$ and is stored each user's smart card where $i=1,\ldots,n$.

2. The TC calculates

\[Conf_i \oplus SI_i = SIc_i\]  \hspace{1cm} (11)

and distributes it to $U_i$ ($i=1,\ldots,n$ : to each user). Note that “$\oplus$” denotes “exclusive OR”.

3. Assume that the group $G$ is composed of $U_i,U_j,U_k$. When $U_i$ share a group communication key with $U_j$ and $U_k$, at first, he inputs his $SIc_i$ and $ID_j,ID_k$ and $m$ which is necessary key length into his smart card, then his card calculates

\[Conf_i \oplus SIc_i = Conf_i \oplus (Conf_i \oplus SI_i) = SI_i\]  \hspace{1cm} (12)

4. Then, it follows [U_i's procedure] 2,3,4,5,6,7 in Section 3.2.

This procedure is shown in Fig 4. $U_j,U_k$ can generate $GK'_G$ by following above procedure as well.

Each $SIc$ is much longer than $X$ and $Nc$, because it involves a large random number “$Conf$” inside. And each $Conf$ is hidden in each user’s smart card secretly, hence, it cannot be reveal to him. Consequently, he can not find out his $SI$ directly.

Instead of $SIc_i = SI \oplus Conf_i$, the TC can set $SIc_i = SI \times Conf_i$. In this case, when a user inputs $SIc_i$ into his smart card his card calculate $SIc_i / Conf_i = SI$.

Introducing $Conf$ realizes that the TC can set length of $X$, $Nc$ shorter, thus, $SI$ can be shorter and it helps reducing the execution time of the modular exponentiation part.

B: Using a computer which has more computation power and memory capacity

We describe a method for executing the modular exponentiation part using a computer which has more computation power and memory capacity than that of smart cards. And also, this method does not reveal the TC's secret informaions, $Nc,X$ to any user.

1. The TC generates large positive prime numbers $R$ and $Nc$ which are different from each other(assume that $R$ and $Nc$ are 256 bits long).

And also, it generates $X$ which is a primitive element over $GF(R)$ and $GF(Nc)$. $SI$(secret information of user) and $Pieces(K_0)$ are defined and generated following the same way in Section 3.1. (note that TC uses $Nc$ and $X$ which are defined here to generate each $SI$).

2. The TC keeps $X,Nc$ and $R$ secret. Also it hides $Pieces(K_0)$ and $Nc$ in each user’s smart card secretly and distributes each $SI$ to each user securely. These distribution of informations are following the same way in Section 3.1. The TC makes $NcR = Nc \times R$ public to all users.

3. Assume that the group $G$ is composed of $U_i,U_j,U_k$. When $U_i$ share a group communication key with $U_j$, $U_k$, at first, he calculates $GK'_G = (SIc_i)^{ID_j,ID_k} \mod Nc$ using a computer which has more computation power and memory capacity than that of his smart card.

4. He inputs $ID_j,ID_k,GK'_G$ and $m$(needed key length) into his smart card.

5. His card follows [U_i's procedure] 2,3 in Section 3.2.

6. And calculates

\[f1(GK'_G, 1) = GK''_G \mod Nc = SIc_i^{ID_j,ID_k} \mod Nc = GK''_G.\]  \hspace{1cm} (13)

7. His card follows [U_i’s procedure] 5,6,7 in Section 3.2

This method is shown in Fig 5. $NcR = Nc \times R$ is made public, but $Nc$ and $R$ must be kept secret and must not be revealed to any user.

In order to find out $Nc$ from $NcR$, a user must factor $NcR$. However $NcR$ is more than 512 bits long and it is quite hard to factor $NcR$ into $Nc$ and $R$. Namely, $Nc$ is kept secret based on the difficulty of factoring “hard” large numbers. This property is used in the RSA cryptosystem[12].

Figure 4: Using a Confounder

This method uses computer which has more computation power and capacity of memory than that of smart cards in order to reduce the execution time of the modular exponentiation part without revealing $X,Nc,R$ to any user.
Figure 5: Using a computer which has more power

$f_{MCK}()$ is an appropriate one-way function in order to generate $G_K_{MCK}$ in Fig 3 and it should not be time-consuming one. If the MCKS is almost completely secure against tampering, this function be simple and Pieces can be small as well.

5 Conclusion

We have presented a 1D-based non-interactive group communication key sharing scheme using smart cards. Our scheme uses smart cards as devices which are secure for tampering. In Section 4.4, considering current power of smart cards, we proposed two methods to shorten execution time of the modular exponentiation part. In the future, smart cards will provide more capacity to calculate and memorize. Once users share a group communication key, they can apply it to any symmetric cryptosystem (e.g., [13][14]).

Smart cards are popularized as compact devices which memorize and calculate important information, because they are excellent in controlling these informations inside. Hence, we believe that it is effective to use smart cards in order to realize a non-interactive group communication key sharing scheme.

As further studies, we will construct an appropriate function $f_{MCK}()$ and length of Pieces considering capacity to calculate and memorize of current smart cards and security.

References