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 conquest 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 IDNIKS, 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.

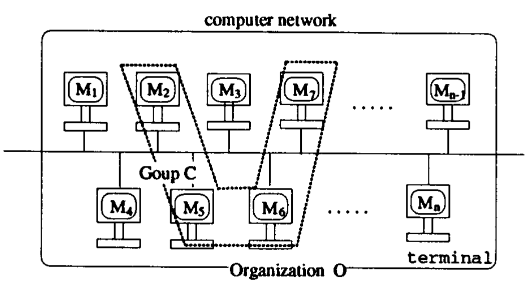
2 The Modified Copy Key(MCK) method

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.



Organization O is composed of $M_1, M_2, \ldots, M_n$
Goup C is consist of M_2, M_5, M_6, M_7

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, M_2 and M_5 are removed from group C, a renewed key must be re-distributed to the rest of members of C, M_6 and M_7. 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* $P_1, P_2, ... P_n$ which are informations for generation of group communication keys (i.e., key generators). And 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 M_i and another member M_j share $n-2$ unique *Pieces*. Note that a *Piece* is a positive integer, and $P_i \neq P_j$ if $i \neq j$. 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.

In Fig 2 the members of a group M_1, M_2, M_4 have P_3, P_5 and P_6, on the other hand, the non-members M_3, M_5, M_6 do not have all of P_3,P_5 and P_6. Since M_1, M_2 and M_4 can generate a key from the common key K_0 and *Pieces* P_3, P_5, P_6, 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 K_0 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.

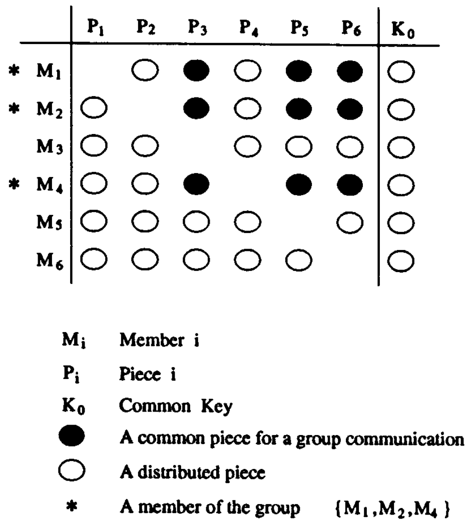


Figure 2: distribution of Pieces

A plain text is encrypted and decrypted by the group communication key. The group communication key K can be generated from a common key K_0 and a *Piece* P as follows:

$$K = f_{MCK}(K_0, P) \quad (1)$$

where $f_{MCK}()$ is an appropriate encryption function which generate a unique group communication key from P and K_0. When multiple $P\{P_1, P_2, ...P_i\}$ are used to generate a group communication key K, it is generated as follows:

$$K_{123...i} = f_{MCK}(...f_{MCK}(f_{MCK}(K_0, P_1), P_2), ..., P_i) \quad (2)$$

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:

$$\sum_{r=1}^{n-1} n-1Cr = 2^{n-1} - 1 \quad (3)$$

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 P_1 and P_2 gets K_{123}, he might be able to get P_3 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 K_0 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.

definition

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

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 $f1, f2, f3$, 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 $MCKS$ choices proper *Pieces* for the group G which is composed of U_i, U_j, U_k. Then it generates K_{ijk}=GK_{MCK_G} by following (1)(2) in Section 2.2.2, using a function f_{MCK}, K_0 and choiced *Pieces*. These are following the MCK method.
3. The $MCKS$ outputs GK_{MCK_G}.

   We assume that the $f_{MCK}()$ 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:

$$P_j \ : \ P_1, P_2, ...P_{i-1}, P_{i+1}, ..., P_n$$
$$[\mathrm{P_j \text{ is a positive integer where } j = 1...n\,, j \neq i}] \tag{4}$$

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 $ID_k(ID$s) 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 GK_{MCK_G} 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 $f_{MCK}()$ be the same function which is described in Section 2.2, GK_{MCK_G} is described as follows :
$$GK_{MCK_G} = f_{MCK}(...f_{MCK}(f_{MCK}(K_0, P_1), P_2)..., P_m) \tag{5}$$
   These procedures are following the MCK method in Section 2.2.
4. The function f1 calculates
$$f1(SI_i, IDs) = SI_i^{ID_j ID_k} modNc = X^{ID_i ID_j ID_k} modNc = GI_G \tag{6}$$
   (a Group Information for group G and subscript G indicates group G).
   Nc is hidden in the f1 as a parameter.
5. The function f2 calculates
$$f2(GI_G, GK_{MCK_G}) = GI_G \bmod GK_{MCK_G} = GK_G \neq 0 \tag{7}$$
   (a Group communication Key for G, the subscript G indicates group G)
6. A function f3 is a hash function which reduces GK_G into m bits. That is,
$$f3(GK_G, m) = GK'_G \tag{8}$$
7. U_i's smart card outputs GK'_G.

We assume that Nc is hidden in each card securely(actually hidden in the f1 as a parameter of it). Note that Nc is larger than GK_{MCK_G}.

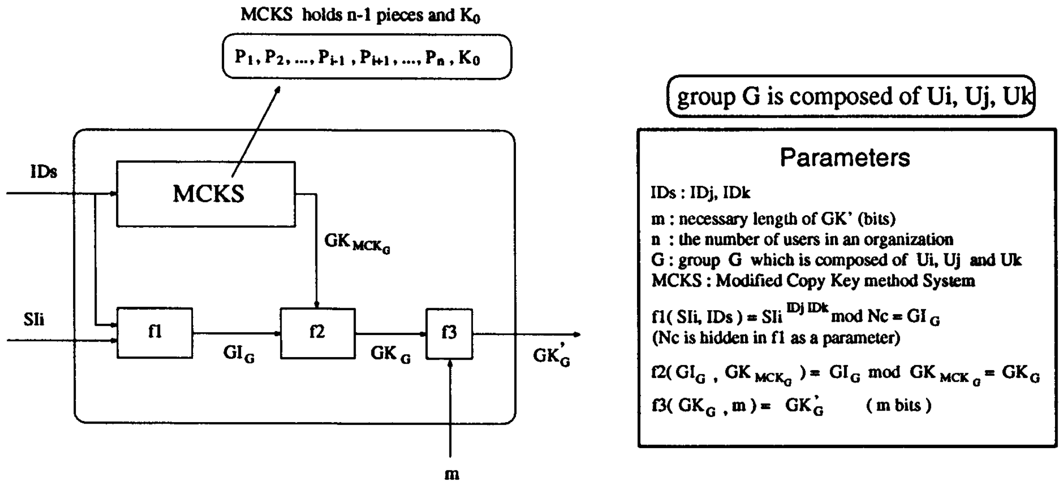


Figure 3: Proposed scheme (U_i's smart card)

[U_j's procedure]
U_j follows U_i's procedure as well. Note that he must replace the index "i" of "j" .

[U_k's procedure]
U_k follows U_i's procedure as well. Note that he must replace the index "i" of "k".

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

$$\begin{aligned} GK_G &= GI_G \bmod GK_{MCK_G} \\ &= (X^{ID_i ID_j ID_k} \bmod Nc) \bmod GK_{MCK_G} \\ &\quad [Nc\, is\, larger\, than\, GK_G]. \end{aligned} \tag{9}$$

$$GK'_G = f3(GK_G, m) \tag{10}$$

Only U_i, U_j, U_k can generate $GI_G = X^{ID_i ID_j ID_k} \bmod Nc$ from their SIs and IDs. And only they can generate GK_{MCK_G} by following the MCK method.

Therefore, any U_l where $l \neq i, j, k$ can not generate GK_{MCK_G} and GI_G, thus he can not generate $GK_G(GK'_G)$. 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

GK is calculated from GI and GK_{MCK}. GI is generated from SI and IDs where SI involves X and Nc. GK_{MCK} 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 GK'_G. In order to make GK'_G, he must generate GI_G and GK_{MCK_G}. If size of X and Nc are large enough, it is quite difficult for U_m to analyze and know X and Nc by himself. Namely, he must find a pair of X and Nc which satisfy $SI_m = X^{ID_m} \bmod Nc$. 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 find 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 $f1$ 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 GK_{MCK_G}.

Thus, they have to analyze the $MCKS$ in order to generate GK_{MCK_G} which is the generator of GK_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, Nc, and all *Pieces*. Therefore smart cards must have structure which is secure for tampering by users.

In order to keep a group communication key $GK'(orGK)$ secure, GI and GK_{MCK} must be kept secure. In order to keep GI secure, the trusted center TC should renews Nc which is hidden in the function $f1$ as a parameter in smart cards of all users, and SI which involves a secret information of the TC "X". Also the TC should renews 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 Nc 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 GI securer, as a result, it makes GK securer.

When TC renews Nc 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 securer.

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, Nc, and $Pieces(K_0)$. Thus, renewal of key generators($X, Nc, Pieces$) is very important to keep security of group communication keys. Since the TC can not renew *Pieces* and Nc 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 Nc 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 = 7days and Ts=1 year.

4.4 Calculation

In our scheme, a smart card must calculate $SI^{ID_1 \dots ID_n} \bmod Nc$, 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, $512bits \times 512bits \bmod 512bits$. 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 choice effective length of X and Nc for modular multiplication(exponentiation), considering security and the execution time.

Considering above state, in order to shorten the execution(calculation) time of this routine, the TC set length of X, Nc, 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 Nc 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 Nc 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., Nc,X,SIs,$Pieces(K_0)$). In addition to them, it generates a large random number "$Confounder_i(Conf_i)$" for a user U_i, whose length is much longer than X and Nc (i=1,...,n, where n is the number of users in an organization). (For example, X and Nc 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,...,n .

2. The TC calculates

$$Conf_i \oplus SI_i = SIc_i \tag{11}$$

and distributes it to U_i (i=1,...,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

$$\mathrm{Conf_i} \oplus \mathrm{SIc_i} = \mathrm{Conf_i} \oplus (\mathrm{Conf_i} \oplus \mathrm{SI_i}) = \mathrm{SI_i} \tag{12}$$

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

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 informations, 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 GI'_G=$(SI_i)^{ID_j ID_k}$ mod Ncr **using a computer which has more computation power and memory capacity than that of his smart card.**

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

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

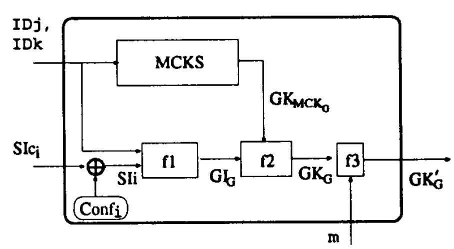
6. And calculates

$$\begin{aligned} &f1(GI'_G, 1) = GI'_G \bmod Nc \\ &= {SI_i}^{ID_j ID_k} \bmod Nc = GI_G. \end{aligned} \tag{13}$$

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 512bits 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].



$\oplus$:XOR Confi: hidden in Ui's smart card

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.

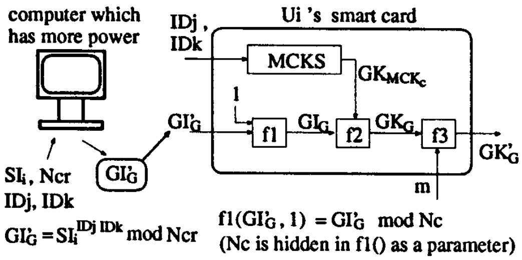


Figure 5: Using a computer which has more power

$f_{MCK}()$ is an appropriate one-way function in order to generate GK_{MCK_G} 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 ID-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

[1] Kum-Yew Lai and Thomas W.Malone, "Object Lens : A Spreadsheet for Cooperative Work", Proc. CSCW '88, 1988.

[2] S.Ichimura, N.Matsuura, K.Okada and Y.Matsushita, "I-CEM : an Intelligent Communication System for Collaborative Work", Proc. 1st International Conference on Parallel and Distributed Information System, December, 1991.

[3] IBM Corp. "Cryptographic key distribution method", IBM Tech. Disclosure Bull, 29[2], pp580-582, 1986.

[4] H.Nakamura, K.Takagi, K.Okada, and Y.Matsushita, "Hierarchical Group Oriented Key Management Method HGK", Computer Security Applications Conference, Dec., 1990.

[5] H.Nakamura, K.Takagi, K.Okada, and Y.Matsushita, "A Group Oriented Key Management Method-GCK", IPS, Japan JWCC 5th, Jul. 1990.

[6] Y.Mutoh, K.Takagi, K.Okada, Y.Matsushita, "The Group Oriented Key Management And Authentication Method", 12th IFIP, World Computer Congress.

[7] W.Diffie, M.E.Hellman, "New Directions in Cryptgraphy", *IEEE Transaction on Information Theory*, vol.IT-22, No.6, Nov., 1976.

[8] H.Tanaka, "Identity-Based Non-interactive Key Sharing", IEICE Trans. Fundamentals, Vol.E77-A , No.1 January 1994.

[9] S.Tujii, T.Itoh and K.Kurosawa "ID-based cryptosystems using discrete logarithm problem", Electronics Letters, pp.1318-1320, 1987.

[10] H.Tanaka, "Identity-based non-interactive key sharing", Proceedings of the 1993 Symposium on Cryptgraphy and Information Security, SCIS93-17C, Jan 1993.

[11] K.Koyama, "Secure Conference Key Distribution Schemes for Conspiracy Attacks", ISEC91-61, March 1992.

[12] Rivest,R.L. ,Shamir, A.Adelman, L., "A method for obtaining digital signatures and publickey cryptosystems", Commun. of the ACM, Vol.21, No.2, pp. 120-126, 1978.

[13] "Data Encryption Standard", FIPS PUB 46, National Bureau of Standards, Washington, D.C., 1977.

[14] S.Miyaguchi, S.Kurihara, K.Ohta and H.Morita, "Expansion of FEAL Cipher", NTT Review, Vol.2, No.6 (1990).

[15] Judy H.Moore "Protocol Failures in Cryptosystems", Proc. of The IEEE, vol 76, No 5, May 1988, pp.594-602.

[16] C.H.Yang and H.Morita, "An Efficient Modular-Multiplication Algorithm for Smart-Card Softeare Implementation", IEICE Japan, Technical Report, ISEC91-58, January 1992.

[17] H.Morita, and C.Huang YANG, "A Modular-Multiplication Algorithm Using Lookahead Determination", IEICE Trans. Fundamentals, Vol.E 76-A, No1 January, 1993.

[18] H.Morita, "A Fast Modular Multiplication Module for Smart Cards", Advances in Cryptology-AUSCRYPT'90, pp. 406-409, Springer-Verlag, 1990.

[19] Hans-Peter Königs "Cryptographic Identification Methods for Smart Cards in the Process of Standardization", IEEE Communications Magazine, pp42-48, June 1991.