

# A Conference Key Distribution Scheme Using Interpolating Polynomials

Chin-Chen Chang\*, Chu-Hsing Lin\*\* and Chien-Yuan Chen\*\*\*

\* *Department of Information Engineering and Computer Science,  
Feng Chia University, 407 Taichung, Taiwan  
ccc@cs.ccu.edu.tw*

\*\* *Department of Computer Science and Information Engineering,  
Tunghai University, 407 Taichung, Taiwan  
chlin@thu.edu.tw*

\*\*\* *Department of Information Engineering,  
No.1, Sec. 1, Syuecheng Rd., Dashu Township, 840 Kaohsiung County, Taiwan  
cychen@isu.edu.tw*

## Abstract

*Conference keys are secret keys used by a group of users commonly with which they can encipher (or decipher) messages such that communications are secure. Based on Diffie and Hellman's PKDS, a conference key distribution scheme is presented in this paper. A sealed lock is used to lock the conference key in such a way that only the private keys of the invited members are matched. Then the sealed lock is thus made public or distributed to all the users, only legitimate users can disclose it and obtain the conference key. In our scheme, the construction of a sealed lock is simple and the revelation of a conference key is efficient as well.*

**Keywords and phrases:** public key distribution, Diffie-Hellman's scheme, conference key, sealed lock

## 1. Introduction

In the age of computers and communications people in different places far away from each other can have a secure conference just by sitting in the front of their own computers via the Internet. A common key, called a conference key, is used to encrypt and decrypt messages which communicate among members participating in the conference. Before a conference is to be held, a conference key has to be generated and distributed safely to members in the conference. The main problem is how this conference key is packed and distributed in such a way that only the legitimate (invited) members can disclose it.

In this paper, we propose a conference key distribution system suitable for broadcast channel. A broadcast channel is characterized that a single transmission from a source user may be received simultaneously by many destination users. The concept of locking, called a sealed lock [15], is used to lock a secret conference key, from which only legitimate users can open it. There is no constraint on the structure of user stations in our system. Moreover, the proposed system has the following properties. First, for a subgroup of users, only one common secret key is required. Second, the conference key can be changed randomly without changing a ciphering key of any user.

The proposed scheme is based on Diffie and Hellman's PKDS [5]. The construction of a sealed lock is straightforward and the revelation of a conference key is simple. In Section 2, we present a brief review of a conference key distribution scheme. Section 3 will describe the overview of our approach and give an example. In Section 4, we analyze the security of the proposed scheme. According to our analysis, the conference key distribution scheme is presented in Section 5. Finally, we have a conclusion.

## 2. Conference Key Distribution

Diffie and Hellman proposed a public key distribution system (PKDS) based on the one-way function  $F(X)=Z^X \bmod p$ , where  $p$  is a large prime number and  $Z$  is a primitive element in Galois field  $GF(p)$ . Here a one-way function means that there exists a fast algorithm for computing  $F(X)$  from any given  $X$ ;

however, the computation of  $X$  from  $F(X)$  is infeasible within a reasonable time limitation [4]. Their PKDS works as follows. Users A and B choose randomly the integers  $X_a$  and  $X_b$ , respectively, from numbers in the range  $[1, p-1]$ . Users A and B keep secretly  $X_a$  and  $X_b$  and compute the corresponding public keys  $Y_a$  and  $Y_b$

$$\begin{aligned} Y_a &= (Z)^{X_a} \text{ mod } p, \text{ and} \\ Y_b &= (Z)^{X_b} \text{ mod } p. \end{aligned} \quad (2.1)$$

$Y_a$  and  $Y_b$  are placed in a public directory or interchanged between users A and B. Then users A and B can compute their common secret key  $K_{ab}$  and follows:

$$\begin{aligned} K_{ab} &= (Y_b)^{X_a} \text{ mod } p, \\ &= (Z)^{X_b X_a} \text{ mod } p, \\ &= (Y_a)^{X_b} \text{ mod } p. \end{aligned} \quad (2.2)$$

This enables users A and B to communicate using encrypted messages by applying any cryptosystem with the key  $K_{ab}$ .

We can see that it is very straightforward to compute the common key  $K_{ab}$ . Each user needs at most  $\log_2 p$  multiplications over  $GF(p)$ . On the contrary, if user A (or user B) intends to expose the private key  $X_b$  (or  $X_a$ ) of his partner, he has to compute discrete logarithms. From the result of Pohlig and Hellman [17], computing discrete logarithms over  $GF(p)$  is considered to be a rather difficult problem if  $p-1$  has at least one large prime factor. Therefore, Eq(2.1) is a one-way function on which the PKDS based.

However, PKDS can serve only for two users to have a session key. If three or more users want to have a conference in common, a conference key is needed, each pair of the users have to keep one secret key. Therefore, in order to communicate with each other among any subgroup of users in the system, we need to derive a common secret key. In addition, for communicating a message to several users, the sender has to perform different encryptions and transmit the ciphertexts several times separately. Clearly, it is very inefficient to use this approach for a conference.

To overcome the above problems, Ingemarsson, Tang, and Wong [8] proposed an elegant scheme named conference key distribution system (CKDS) for any subgroup of  $m$  users to share the same encryption and decryption keys in a network with  $n$  users, where  $2 \leq m \leq n$ . Conditionally, these  $m$  participants users have to be connected in a ring structure first before the progress of work follows. Within the ring structure, each user has to process and transmit the message received from his previous user station. Under this sequential order of message processing  $m-1$  times, and finally the common conference key can be derived. However, an attacker may intercept the message transmitted along the ring. By putting the intercepted

message together, a threat of wiretapping to the keys thus exists.

Generally, the CKDS can be classified into two categories: one is the non-ID-based type [3, 8, 13, 16, 19] and the other is the ID-based type [2, 14, 11, 12]. Unfortunately, most of the published ID-based CKDS are shown to be insecure [11, 12, 18, 20]. Therefore, in this paper, we focus our attention on the non-ID-based CKDS. In the following, we are going to review a practical non-ID-based CKDS [16].

In 1988, Lu, et al. [16] proposed a conference key distribution system based on the Lagrange interpolating polynomial. Let us briefly describe their method as follows. As indicated in Diffie and Hellman's PKDS, each user possesses a private key  $X_i$  and makes the key  $Y_i$  public. Now we assume that there are  $r$  users, namely  $U_1, U_2, \dots, U_r$ , being invited to the conference by the chairman  $U_0$ . First, a conference key  $\alpha$  is chosen by  $U_0$  and  $2r$  numbers are computed, which are  $\{K_{01}, K_{02}, \dots, K_{0r}\}$  and  $\{k'_{01}, k'_{02}, \dots, k'_{0r}\}$ , such that  $K_{0i} = (Y_i)^{X_0} \text{ mod } p = (Y_0)^{X_i} \text{ mod } p$ , for  $1 \leq i \leq r$ .

Secondly,  $U_0$  construct a Lagrange interpolating polynomial  $L(x)$  as follows.

$$L(x) = \sum_{i=1}^r \alpha K'_{0i} \prod_{j=1, j \neq i}^r ((x - K_{0j}) / (K_{0i} - K_{0j})) \text{ mod } p \quad (2.3)$$

In other words,  $L(x)$  is a polynomial with degree  $r-1$  passing the  $r$  points  $(K_{0i}, \alpha K'_{0i})$ ,  $1 \leq i \leq r$ . Then  $L(x)$  is transmitted to users participating in the conference. Now the conference key  $\alpha$  is hidden in  $L(x)$ . Here we also like to point out that from Diffie and Hellman's formula, Eq(2.2), we have  $K_{0i} = (Y_i)^{X_0} \text{ mod } p = (Y_0)^{X_i} \text{ mod } p$ . Therefore, on receiving  $L(x)$ , an invited user  $U_i$  can evaluate the polynomial  $L(K_{0i})$  and would obtain the value  $\alpha K'_{0i}$ ; i.e., he obtain  $L(K_{0i}) = \alpha K'_{0i}$ . Furthermore, he can obtain the conference key by the following.

$$\begin{aligned} \alpha &= (\alpha K'_{0i}) \cdot (K'_{0i})^{-1} \text{ mod } P \\ &= L(K_{0i}) \cdot (K'_{0i})^{-1} \text{ mod } P \end{aligned} \quad (2.4)$$

Where  $(K'_{0i})^{-1}$  indicates the multiplication inverse of  $(K'_{0i})$  with modulus  $p$ . However, each time when a conference is to be held, a Lagrange interpolating polynomial has to be constructed. Moreover, every invited user must evaluate  $L(x)$  to obtain conference key  $\alpha$ .

In the next section, we present a new conference key distribution scheme. By using our scheme, interpolating polynomials are constructed just once and for all.

### 3. Background of Our Scheme

Imagine that there is a group  $G$  containing  $n+1$  users, denoted by  $U_0, U_1, U_2, \dots$ , and  $U_n$ , in a networking system. Let  $G'$  indicate a nonempty subgroup of  $m$  users within  $G$ , where  $1 \leq m \leq n$ . Suppose that, initially, each user  $U_i$  keeps secret a private key  $X_i$ , chosen randomly by  $U_i$  from numbers in the range  $[1, p-1]$ , where  $p$  is a large prime number, and publishes the associated public key  $Y_i = (Z)^{X_i} \bmod p$ , where  $Z$  is a primitive element in the  $GF(p)$ , where  $GF(p)$  indicates the Galois field over  $p$ . Without loss of generality, assume that  $U_0$  is the chairman and  $U_1, U_2, \dots, U_r$  are users invited to the conference; i.e.,  $G' = \{U_1, U_2, \dots, U_r\}$ . In order to hold a secure conference among the users in  $G'$ , a secret conference key, denoted by  $\alpha$ , has to be created by the chairman for the conference. Note that  $\alpha$  is also chosen in  $GF(p)$ .

We can see that if there is a secure method which can conceal the conference key  $\alpha$  then the corns of the conference key distribution system can be solved. Since the conference key is enciphered, only one copy is needed to be sent in a broadcast system. Further, since the conference key is generated when a conference is going to be held, no extra key has to be kept in secret. Based upon these ideas, a new approach is proposed. A lock, called the sealed lock, is created and applied to lock the conference key. Note that the concept of a sealed lock for conference key distribution was proposed by Lin, et al. [15]. The sealed lock only matches the private keys of users in  $G'$ .

Accordingly, we may assume that the conference key is hidden in the sealed lock and the lock satisfies two requirements. First, since only users in  $G'$  are invited, the lock should be opened only by the users in  $G'$ , not any user in  $G-G'$ . Second, the lock should be variant according to different conference key  $\alpha$ . That is, each time we use different lock depending on different conference key. Briefly, a sealed lock has to rest functionally on not only the conference key  $\alpha$  but also the ciphering keys of the users.

Now, the remaining problem is how we can construct the sealed lock. Before presenting the method, let us describe the informal steps of the scheme. First,  $U_0$  chooses a  $n \times n$  nonsingular matrix over  $GF(p)$ . Let the row vectors of  $K$  be  $K_1, K_2, \dots, K_n$ . Let  $B = (b_1, b_2, \dots, b_n)^T$ , where  $b_i$ 's are unknowns to be determined and  $T$  indicates a transpose operation on vectors. Let  $C = (c_1, c_2, \dots, c_n)^T$ , where  $c_i = \alpha$  if user  $U_i$  in  $G'$ ; otherwise,  $c_i = 0$ . Since the  $n$  row vectors of  $K$  are linearly independent, they constitute a basis [6]. Therefore, corresponding to any  $n$ -tuple vector  $C = (c_1,$

$c_2, \dots, c_n)^T$ , a unique coordinate vector  $B = (b_1, b_2, \dots, b_n)^T$ , for representing  $C$  in the basis, can be found by solving the following linear equations:

$$KB = C, \quad (3.1)$$

or equivalently  $B = K^{-1}C$ ,  $K^{-1}$  indicates the inverse matrix of  $K$ . From another point of view, it means that when the coordinate vector  $B$  is obtained, the  $i^{\text{th}}$  component (i.e.,  $c_i$ ) of the vector  $C$  becomes the result of  $K_i * B$ , where  $*$  indicates the vector product in  $GF(p)$ . That is  $K_i * B = c_i = \alpha$ , if  $U_i$  is in  $G'$ ; otherwise  $K_i * B = 0$ .

From the above statements, it is not difficult to see that if the chosen row vector  $K_i$  could be possessed by user  $U_i$  and the vector  $B$  were made public, then each user  $U_i$  would be able to compute the value  $c_i$  by himself (or herself). Thus, the invited users would obtain  $c_i = \alpha$ , the conference key; and the uninvited users would obtain  $c_i = 0$ . However, how can we distribute  $K_i$  to user  $U_i$  securely? In the following, we give a method to conceal the matrix  $K$  in such a way that only user  $U_i$  can reveal the corresponding  $i^{\text{th}}$  row vector  $K_i$ .

First, for each column of the matrix  $K$ , namely column  $j$ , we construct an interpolating polynomial  $F_j$  [1, 9, 10] with degree  $n-1$  passing through the  $n$  points  $(ID_i, (k_{ij})^p \bmod Q)$ ,  $1 \leq i \leq n$ . Here  $ID_i$  indicates the identification number of user  $U_i$  and  $Q = q_1 \times q_2$  is the product of two large prime numbers. Note that as aforementioned we assume that user  $U_0$  is the chairman and only users  $U_1, U_2, \dots$ , and  $U_r$  are invited to the conference. Moreover, for each column of the matrix  $K$ , e.g. the  $j^{\text{th}}$  column, we construct another interpolating polynomial, namely  $H_j$ , with degree  $n-1$  passing through the  $n$  points  $(ID_i, k_{ij}^{(Y_i)^{X_0} \bmod p} \bmod Q)$ ,  $1 \leq i \leq n$ . The construction steps of an interpolating polynomial, one can consult [1, 9, 10]. Therefore, we obtain a set of  $2n$  polynomials, namely  $F = \{F_1, F_2, \dots, F_n, H_1, H_2, \dots, H_n\}$ . Finally, the set  $F$  of polynomials are made public by the chairman to all the users in the system.

Now, when the user  $U_s$ , with identification  $ID_s$ , reads the set  $F$  of polynomials, he (or she) can evaluate the values of polynomials  $F_i(ID_s)$  for  $1 \leq i \leq n$ . We can see that the result will be as indicated below:

$$\begin{aligned} F_1(ID_s) &= k_{s1}^p \bmod Q, \\ F_2(ID_s) &= k_{s2}^p \bmod Q, \\ &\vdots \\ F_n(ID_s) &= k_{sn}^p \bmod Q. \end{aligned} \quad (3.2)$$

Similarly, he can also evaluate the results of polynomials  $H_i(ID_s)$  and has the following equalities

$$\begin{aligned} H_1(ID_s) &= k_{s1}^{(Y_s)^{X_0} \bmod p} \bmod Q, \\ H_2(ID_s) &= k_{s2}^{(Y_s)^{X_0} \bmod p} \bmod Q, \\ &\vdots \\ H_n(ID_s) &= k_{sn}^{(Y_s)^{X_0} \bmod p} \bmod Q, \end{aligned} \quad (3.3)$$

It has  $(Y_s)^{X_0} \bmod p = (Y_0)^{X_s} \bmod p$  and Eq(3.3) becomes:

$$\begin{aligned} H_1(\text{ID}_s) &= k_{s1}^{(Y_0)^{X_s} \bmod p} \bmod Q, \\ H_2(\text{ID}_s) &= k_{s2}^{(Y_0)^{X_s} \bmod p} \bmod Q, \\ &\vdots \\ H_n(\text{ID}_s) &= k_{sn}^{(Y_0)^{X_s} \bmod p} \bmod Q, \end{aligned} \quad (3.4)$$

Further, the key point is that how can the user  $U_s$  deduce the corresponding row vector  $K_s$  by knowing Eq(3.2) and Eq(3.4). The answer will become clear when Theorem 3.1 is proved.

### Theorem 3.1

Given  $b_1, b_2, e_1,$  and  $e_2$  such that  $b_1 = b^{e_1} \bmod n$  and  $b_2 = b^{e_2} \bmod n$ , where  $b < n$ . Then  $b^r \bmod n$  can be easily computed if  $\gcd(e_1, e_2) = r$ .

**Proof.** Since  $\gcd(e_1, e_2) = r$ , from Euclidean algorithm we can find a pair of  $(s_1, s_2)$  such that  $s_1 e_1 + s_2 e_2 = r$ . Therefore, we have

$$\begin{aligned} b^r \bmod n &= b^{(s_1 e_1 + s_2 e_2)} \bmod n \\ &= ((b^{e_1})^{s_1}) \bmod n \cdot ((b^{e_2})^{s_2}) \bmod n \\ &= ((b_1)^{s_1} \bmod n) \cdot ((b_2)^{s_2} \bmod n) \bmod n \end{aligned}$$

Since  $\gcd(p, (Y_0)^{X_s} \bmod p) = 1$ , Theorem 3.1 (by letting  $r=1$ ) can be applied to Eq(3.2) and Eq(3.4) for solving  $K_s = (k_{s1}, k_{s2}, \dots, k_{sn})$ . Moreover, with the obtained  $K_s$  the value  $c_s$  can be derived by computing  $K_s * B = (k_{s1}, k_{s2}, \dots, k_{sn}) * (b_1, b_2, \dots, b_n) = c_s$ . If the user  $U_s$  is in  $G'$  then  $c_s = \alpha$ ; otherwise, he would find that  $c_s = 0$ , no information associated with the conference key is revealed.

It is easy to see that the sealed lock, the vector  $B$ , satisfies the previous two requirements. Here we would like to point out that to open the sealed lock in the conference key distribution system, one's private key, not any extra key, is needed. Moreover, the conference key can be changed in a convenient way within the conference. When a suspected attack is found, the chairman may change the conference key as he wishes.

### Example 3.1

Let the group  $G$  contain four users, denoted by  $U_0, U_1, U_2,$  and  $U_3$ . Let  $p$  be 11 and a primitive element  $Z=2$  in  $\text{GF}(p)$ . Then, each user  $U_i$  keeps secret a private key  $X_i$ , chosen randomly by  $U_i$  from numbers in the range  $[1, p-1]$  and publishes the public keys  $Y_i = Z^{X_i} \bmod p$ . Let  $X_0=8, X_1=6, X_2=9, X_3=4, Y_0=3, Y_1=9, Y_2=6,$  and  $Y_3=5$ . We also let  $\text{ID}_1=1, \text{ID}_2=2,$  and  $\text{ID}_3=3$ . Without loss of generality, assume that  $U_0$  is the chairman and  $U_1$  and  $U_2$  are users being invited to the conference; i.e.,  $G' = \{U_1, U_2\}$ . In order to hold a secret conference key  $\alpha=7$  among the users in  $G'$ , the chairman executes the following steps:

1. The chairman chooses a  $3 \times 3$  nonsingular matrix over  $\text{GF}(11)$  as

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 6 \\ 7 & 5 & 3 \end{bmatrix},$$

and computes the inverse  $K^{-1}$  of  $K$

$$K^{-1} = \begin{bmatrix} 5 & 2 & 2 \\ 8 & 7 & 0 \\ 8 & 2 & 3 \end{bmatrix}.$$

2. The conference key  $\alpha=7$ , he generates the vector  $C = (c_1, c_2, c_3)^T$ , where  $c_i = \alpha$  if  $U_i$  is in  $G'$ ; otherwise,  $c_i = 0$ . So,  $C = (7, 7, 0)$ .
3. The vector  $B$  is generated by

$$B = \begin{bmatrix} 5 & 2 & 2 \\ 8 & 7 & 0 \\ 8 & 2 & 3 \end{bmatrix} \begin{bmatrix} 7 \\ 7 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 4 \end{bmatrix}.$$

(All values are over  $\text{GF}(p)$ )

4. Selecting  $Q=23 \times 29=667$  and  $P=673$ , the chairman can construct six interpolating polynomials  $F_1, F_2, F_3, H_1, H_2,$  and  $H_3$ . According to the construction of the three points  $(\text{ID}_1, k_{11}^P \bmod Q), (\text{ID}_2, k_{21}^P \bmod Q),$  and  $(\text{ID}_3, k_{31}^P \bmod Q)$ , which are  $(1, 1), (2, 47),$  and  $(3, 574)$ , is  $577x^2 - 339x + 436 \pmod{673}$ . Similarly, we have

$$F_2(x) = 554x^2 - 362x + 528 \pmod{673} \text{ and}$$

$$F_3(x) = 650x^2 - 581x + 323 \pmod{673}.$$

In addition, polynomial  $H_i$  can be constructed. The polynomial  $H_1(x)$ , passing through the three points  $(1, 1), (2, 16),$  and  $(3, 400)$ , is  $521x^2 - 202x + 355 \pmod{673}$ . Similarly, we have  $H_2(x) = 652x^2 - 617x + 646 \pmod{673}$  and  $H_3(x) = 98x^2 - 365x - 379 \pmod{673}$ .

5. Publish the set  $F = \{F_1, F_2, F_3, H_1, H_2, H_3\}, Q,$  and  $B$  to all the users in  $G$ .

Each user  $U_i$ , say  $U_2$ , in the group  $G'$  can reveal the conference key as follows:

1. Compute  $F_1(\text{ID}_2) = 47, F_2(\text{ID}_2) = 1,$   
 $F_3(\text{ID}_2) = 415, H_1(\text{ID}_2) = 16, H_2(\text{ID}_2) = 1,$  and  
 $H_3(\text{ID}_2) = 629.$

2. Because  $p (=11)$  is coprime to  $Y_0^{X_2} \bmod p (=4)$ , we have  $(-1) \cdot 11 + 3 \cdot 4 = 1$ . Therefore, the 2<sup>nd</sup> row vector of matrix  $K$ , namely  $K_2 = (k_{21}, k_{22}, k_{23})$ , can be computed by the following expressions:

$$k_{21} = (47)^{-1} (16)^3 \bmod 667 = 2,$$

$$k_{22} = (1)^{-1} (1)^3 \bmod 667 = 1,$$

$$k_{23} = (415)^{-1} (629)^3 \bmod 667 = 6.$$

So,  $K_2 = (2, 1, 6)$ .

3. Reveal the conference key

$$\alpha = K_2 * B \text{ mod } p = [216] \begin{bmatrix} 5 \\ 6 \\ 4 \end{bmatrix} \text{ mod } 11 = 7$$

#### 4. Security Consideration

In this section, we shall discuss the security of the proposed scheme.

In the following, attacks to conference key and the personal private keys, are considered to demonstrate the security of the system. First, an intruder, namely  $U_j$ , not in  $G'$ , may try to find the conference key  $\alpha$ . Since  $U_j$  is not in  $G'$ , he will find  $c_j=0$ . Not any information from the lock he can get about the conference key. Secondly, the intruder may be a member in  $G'$  itself. After knowing the conference key  $\alpha$ , he tries to obtain a private key  $X_s$  of some other user  $U_s$  in  $G'$ . With this private key  $X_s$ , he can decrypt the messages for  $U_s$  from another conference in which he does not participate. Form Eq(3.2) and Eq(3.4), we know that the intruder can compute  $2n$  pairs of numbers  $F_i(ID_s)$  and  $H_i(ID_s)$ ,  $1 \leq i \leq n$ . By knowing these  $2n$  pairs, to solve each  $k_{si}$  by the Theorem 3.1, the value of  $(Y_0)^{X_s} \text{ mod } p$  has to be known in advance. On the other hand, if the value of  $(Y_0)^{X_s} \text{ mod } p$  is known, it has still to face the discrete logarithm problem to get  $X_s$ . However, computing discrete logarithm has been seen as a difficult problem as aforementioned.

It will be not difficult to see that the chairman only has to publish the set  $F$  of interpolating polynomials to all the users just once and for all. On the other hand, a legitimate user can obtain the row vector corresponding to him by using his ID number and his private key. Further the conference key can thus be computed easily. Moreover, when different subgroup of users in group  $G$  are invited to hold a different conference, all what the chairman has to do is compute a new vector  $B$  and publish it. With this new vector  $B$  and the previously published set  $F$ , a new conference for another subgroup of users can be started. Besides, if unfortunately the conference key  $\alpha$  is suspected to be under attack. The chairman can compute a new conference key  $\alpha'$  just by replacing the old vector  $B$  with a new vector  $B'$  within the same conference without any modification to the set  $F$ . Nevertheless, one disadvantage is that the original matrix  $K^{-1}$  has to be kept secretly by the chairman. For security consideration, we suggest that matrix  $K^{-1}$  would be discarded after being transformed to the following two matrices.

$$\hat{K} = \begin{bmatrix} k_{11}^p \text{ mod } Q & k_{12}^p \text{ mod } Q & \dots & k_{1n}^p \text{ mod } Q \\ k_{21}^p \text{ mod } Q & k_{22}^p \text{ mod } Q & \dots & k_{2n}^p \text{ mod } Q \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1}^p \text{ mod } Q & k_{n2}^p \text{ mod } Q & \dots & k_{nn}^p \text{ mod } Q \end{bmatrix}$$

$$\tilde{K} = \begin{bmatrix} (k_{11})^{s_1} \text{ mod } Q & (k_{12})^{s_1} \text{ mod } Q & \dots & (k_{1n})^{s_1} \text{ mod } Q \\ (k_{21})^{s_2} \text{ mod } Q & (k_{22})^{s_2} \text{ mod } Q & \dots & (k_{2n})^{s_2} \text{ mod } Q \\ \vdots & \vdots & \ddots & \vdots \\ (k_{n1})^{s_n} \text{ mod } Q & (k_{n2})^{s_n} \text{ mod } Q & \dots & (k_{nn})^{s_n} \text{ mod } Q \end{bmatrix}$$

for some integer  $s_1, s_2, \dots, s_n$  in  $GF(p)$ . When needed, the chairman can reconstruct the original matrix  $K$  from  $\hat{K}$  and  $\tilde{K}$  in the same way.

Unfortunately, this system is vulnerable to the following two attacks. First attack works as follows. Assume that it is not difficult to find out whether user  $U_i$  participated in a conference or not. If user  $U_i$  is not a legal participant in the conference  $j$ , the intruder can store  $B_j$  sent by the chairman and know that the corresponding  $c_j$  is 0. He continues to store such  $B$ 's until he obtains a subset  $\{B_{j_1}, B_{j_2}, \dots, B_{j_{n-1}}\}$  of these  $B$ 's that form a matrix of rank  $n-1$ . If the intruder is ever involved in a conference, say the  $j_n^{\text{th}}$  conference, with user  $U_i$ , he will know  $\alpha_{j_n}$  and the  $B_{j_n}$ . Now he has a matrix  $\tilde{B} = [B_{j_1}, B_{j_2}, \dots, B_{j_n}]$ . Because the system uses the same matrix  $K$  over and over again, the intruder can recover row  $K_i$  by solving the equation  $K_i * \tilde{B} = [0, 0, \dots, \alpha_{j_n}]$ . Using  $K_i$ , the intruder can find all conference keys for conferences in which user  $U_i$  participated. Second attack like first one is also to recover row  $K_i$  of the user  $U_i$ . If the intruder is ever involved conferences  $j_1, j_2, \dots, j_n$  with user  $U_i$ , he will know  $\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_n}$  and  $B_{j_1}, B_{j_2}, \dots, B_{j_n}$  sent for these conference. Now he has a matrix  $\tilde{B} = [B_{j_1}, B_{j_2}, \dots, B_{j_n}]$ . Because the system uses the same matrix  $K$ , the intruder can recover row  $K_i$  by solving the equation  $K_i * \tilde{B} = [\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_n}]$ .

To eliminate these attacks, we have the detail of our proposed system in the next section.

#### 5. Secure Conference Key Distribution

Let each user  $U_i$  have a private key  $X_i$  and a public key  $Y_i$  as indicated in Diffie and Hellman's PKDS. Now, if a nonempty subgroup  $G'$  of users, with some user  $U_0$  as the chairman, will hold a conference

securely. We assume that a  $n \times n$  nonsingular matrix  $K$  over  $GF(p)$  has been generated by chairman. First, the user  $U_0$  selects a secret conference key  $\alpha$  which will be used to encrypt and decrypt message among users in  $G'$ . To avoid two attacks mentioned in Section 4, let  $C$  be  $(c_1, c_2, \dots, c_n)^T$ , where  $c_i = \alpha((Y_i)^{X_0} \bmod p)$  if  $U_i$  is in  $G'$ ; otherwise,  $c_i$  is a random number such that  $c_i \neq \alpha((Y_i)^{X_0} \bmod p)$ . Then the user  $U_0$  computes a vector  $B$  by solving Eq(3.1) as mentioned previously and makes it public to all the users in the system. Further,  $U_0$  computes and publishes a set of  $2n$  interpolating polynomials  $F = \{F_1, F_2, \dots, F_n, H_1, H_2, \dots, H_n\}$  in which the matrix  $K$  is concealed. On the other hand, on receiving of  $F$ , the user  $U_s$  should be able to compute  $2n$  pairs of numbers  $F_j(ID_s)$  and  $H_i(ID_s)$ ,  $1 \leq i \leq n$ . If the users  $U_s$  is in  $G'$ , an invited user, then by applying Theorem 3.1 he (or she) is capable of obtaining the row vector  $K_s$  corresponding him (or her). Therefore, by  $K_s$  and  $B$ ,  $U_s$  can compute the conference key by  $(K_s * B)((Y_i)^{X_0} \bmod p)^{-1} = c_s((Y_i)^{X_0} \bmod p)^{-1} = \alpha$ . Otherwise, when the user  $U_s$  is not in  $G'$ , he can only compute  $(K_s * B)((Y_i)^{X_0} \bmod p)^{-1} = c_s((Y_i)^{X_0} \bmod p)^{-1} \neq \alpha$ .

In the following, let us state formally the algorithm for the conference key distribution scheme.

#### Algorithm 5.1: System Generation for Chairman

Input: Identification numbers  $ID_i$ 's and the public key  $Y_i$  for users in  $G'$  and a prime number  $p$ .

Output: A set  $F$  of  $2n$  interpolating polynomials and a number  $Q$ .

Step1: [Construct an  $n \times n$  matrix]

Construct a nonsingular matrix  $K = [k_{ij}]_{n \times n}$ .

Step2: [Construct  $n$  interpolating polynomials  $F_j$ 's]

For  $j = 1$  to  $n$

Compute  $F_j$  passing through the  $n$  points  $(ID_v, (k_{vj}^p \bmod Q))$ 's,  $1 \leq v \leq n$  and  $Q = q_1 q_2$ , the product of two large primes.

Next  $j$ .

Step3: [Construct  $n$  interpolating polynomials  $H_j$ 's]

For  $j = 1$  to  $n$

Compute  $F_j$  passing through the  $n$  points  $(ID_v, (k_{vj}^{(Y_i)^{X_0} \bmod p})) \bmod Q$ ,  $1 \leq v \leq n$  and  $Q = q_1 q_2$

Next  $j$ .

Step4: [Distribute the polynomials]

Publish the set  $F = \{F_1, F_2, \dots, F_n, H_1, H_2, \dots, H_n\}$  and  $Q$  to all users in  $G$ .

#### Algorithm 5.2: Constructing a Sealed Lock

Input: The nonsingular matrix  $K$  and a subgroup of users.

Output: A sealed lock  $B$ .

Step1: [Select a conference key for users in  $G'$ ]

Select a conference key  $\alpha$ . Let vector  $C = (c_1, c_2, \dots, c_n)^T$ , where  $c_i = \alpha((Y_i)^{X_0} \bmod p)$  if  $U_i$  is in subgroup; otherwise,  $c_i$  is a random number such that  $c_i \neq \alpha((Y_i)^{X_0} \bmod p)$ .

Step2: [Find a vector  $B$ ]

Compute the vector  $B = K^{-1}C$ . Let  $B = (b_1, b_2, \dots, b_n)^T$  and publish it.

Step3: [Distribute the sealed lock]

Distribute  $B$  to all the users in the system.

Note that as indicated in the above algorithm, the vector  $B$  is used as sealed lock in which the conference key  $\alpha$  is hidden. Now for any user  $U_i$  in  $G'$ , he can reveal the conference key from the set  $F$  and the vector  $B$  by using his own private key  $X_i$ . The revealing procedure is described as follows.

#### Algorithm 5.3: Revelation of the conference key

Input: The set  $F$ , vector  $B$ ,  $ID_s$ ,  $Y_0$ ,  $p$ ,  $Q$ , and the private key  $X_s$  of the user  $U_s$ .

Output: The conference key  $\alpha$ .

Step1: [Evaluate polynomial  $F_j$ 's]

For  $j = 1$  to  $n$

Compute  $F_j(ID_s)$ .

Next  $j$ .

Step2: [Evaluate polynomial  $H_j$ 's]

For  $j = 1$  to  $n$

Compute  $H_j(ID_s)$ .

Next  $j$ .

Step3: [Obtain the  $s^{\text{th}}$  row vector  $K_s$ ]

For  $i = 1$  to  $n$

Compute the  $i^{\text{th}}$  component of  $K_s$ , namely  $k_{si}$ , by Theorem 3.1.

Next  $i$ .

Step4: [Reveal the conference key]

Compute the conference key by

$$\alpha = (K_s * B)((Y_0)^{X_s} \bmod p)^{-1}.$$

After all the users in  $G'$  obtain the common secret key  $\alpha$ , one can communicate with the others as he wishes. Messages are encrypted and decrypted by using the key  $\alpha$  and a conference will proceed securely among the users. The key  $\alpha$  may be generated at the beginning of the conference by the chairman and discarded when the conference is closed or it may be changed randomly within the period of time of this conference. From the above algorithm, in order to compute  $\alpha$ , a user  $U_s$  in  $G'$  only has to reveal the vector  $K_s$  by using his own private key  $X_s$ . Therefore, for conferences among users in any nonempty subgroup of  $G$ , the private keys needed to be kept by each user in the system are still the same.

## 6. Conclusions

It can be foreseen that teleconferencing will play a more and more important role in the age of computers and communications. However, the key issue is how we can design a convenient and secure way for conferencing by using our computers and communication networks. In this paper, we have proposed a method to computer a sealed lock by communicating the encrypted messages among users in the computer networks. By using the sealed lock, secure distribution of a conference key to all the station nodes is feasible in the network systems.

**Acknowledgement:** This research is partially supported by the National Science Council, Taiwan, by contract number: NSC-95-2221-E-029-020-MY3

## References

- [1] Burden, R. L., J. D. Faires, and A. C. Reynolds, Numerical Analysis, Second Edition, Prindle, Weber & Schmidt, Reading, Massachusetts, 1981.
- [2] Chikazawa, T. and A. Yamagishi, "An Improved Identity-Based One-Way Conference Key Sharing System," Communications on the move. Singapore. ICCS/ISITA'92, 1992, pp. 270-273.
- [3] Chiou, G. H. and W. T. Chen, "Secure Broadcasting Using the Secure Lock," IEEE Trans. on Software Engineering, Vol. 15, No. 8, Aug. 1989, pp. 929-934.
- [4] Denning, D. E. R., Cryptography and Data Security, Reading, Mass., Addison-Wesley, 1982.
- [5] Diffie W. and M. E. Hellman, "New Directions in Cryptography," IEEE Trans. on Information Theory, Vol. IT-22, Nov. 1976, pp. 644-654.
- [6] Farkas, I. and M. Farkas, Introduction to Linear Algebra, Adam Hilger Ltd., 1975.
- [7] Hwang, T. and J. L. Chen, "Identity-Based Conference Key Broadcast Systems," IEE Proceedings Computers and Digital Techniques, Vol. 141, Jan. 1994, pp. 57-60.
- [8] Ingemarsson, I., D. T. Tang, and C. K. Wong, "A Conference Key Distribution System," IEEE Trans. on Information Theory, Vol. IT-28, No. 5, Sep. 1982, pp. 714-720.
- [9] Jain, M. K., S. R. K. Lyengar, and R. K. Jain, Numerical Methods for Scientific and Engineering Computation, Wiley Eastern Limited, Reading, New Delhi, 1985.
- [10] Knuth, D. E., The Art of Computer Programming, Vol. 2: Seminumerical algorithms, 2nd edition, Addison-Wesley, Reading, Mass., 1980.
- [11] Koyama, K. and K. Ohta, "Identity-Based Conference Key Distribution Systems," Advance in Cryptology, Eurocrypt'87, 1987, pp. 175-184.
- [12] Koyama, K. and K. Ohta, "Security of Improved Identity-Based Conference Key Distribution Systems," Advance in Cryptology, Eurocrypt'88, 1988, pp. 11-19.
- [13] Laih, C. S., L. Harn, and J. Y. Lee, "A New Threshold Scheme and its Application on Designing the Conference Key Distribution Cryptosystem," Information Processing Letters, Vol. 32, No. 3, 1989, pp. 95-99.
- [14] Laih, C. S. and S. M. Yen, "On the Design of Conference Key Distribution System for the Broadcasting Networks," Proceedings IEEE INFOCOM'93, 12<sup>nd</sup> Annual Joint Conference of IEEE Computer and Communication Society, 1993, pp. 11d.1.1-11d.1.8.
- [15] Lin, C. H., C. C. Chang and R. C. T. Lee, "A Conference Key Broadcasting System Using Sealed Lock," Information Systems, Vol. 17, No. 4, 1992, pp. 323-328.
- [16] Lu, E. H., W. Y. Hwang, L. Harn and J. Y. Lee, "A Conference Key Distribution System Based On The Lagrange Interpolating Polynomial," Proceedings IEEE INFOCOM'88, 7<sup>th</sup> Annual Joint Conference of IEEE Computer and Communication Society, 1988, pp. 1092-1094.
- [17] Pohlig, S. C. and M. E. Hellman, "An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance," IEEE Trans. on Information Theory, Vol. IT-24, Jan. 1978, pp. 106-110.
- [18] Shimbo, A. and S. Kawamura, "Cryptanalysis of several Conference Key Distribution Schemes," Proc. of Asiacypt'91, 1991, pp. 155-160.
- [19] Wu, T. C. and Y. S. Yeh, "A Conference Key Distribution System Based on Cross-product," Computers and Mathematics with Applications, No. 4, 1993, pp.39-46.
- [20] Yacobi, Y., "Attack on the Koyama-Ohta Identity-Based Conference Key Distribution Scheme," Advance in Cryptology, Crypto'87, 1987, pp. 429-433.