

# Conference Key Agreement Protocol Employing a Symmetric Balanced Incomplete Block Design

Okbin Lee

Department of Computer Science,  
Chosun University,  
Kwang – ju Cty, Korea

SeongYeol Kim

Department of Computer Science,  
Ulsan College  
Ulsan City, Korea

**Abstract** - *A conference key agreement system is a scheme to generate a conference key in a contributory manner in order to communicate with each other securely among participants. In this paper an efficient conference key agreement system is proposed by employing symmetric balanced incomplete block design(SBIBD), one class of block designs. The protocol presented not only minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key.*

*Our protocol constructs a conference key which takes the forms of  $\prod_{i=0}^{v-1} R_i$ , where  $v$  is the number of participants and  $R_i$  is a random number generated from member  $i$ . In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is  $O(v\sqrt{v})$ . Our protocol can be proved as computationally difficult to calculate as discrete logarithms.*

**Keywords:** key agreement system

## 1 Introduction

As teleconference is widely adopted in a enterprise environment, security service for the conference is on the rise to provide communication privacy. Conference key can be considered as a good solution to achieve secure communication. A conference key agreement system is a scheme to generate a conference key in a contributory manner in order to communicate with each other securely among participants.

Conference key can be generated by two different types. One is centralized method. In this case, TTP(Trusted Third Party) or single member generates a key and distributes it. Simple but this has significant drawbacks such as overall reliance on a single party. The other is contributory method.

Each group member contributes to generate a key independently. This is a process of key agreement among members. Conference key agreement systems can be classified into two categories according to the group feature whether it is for static group or dynamic group. Conference key agreement for dynamic group needs to consider group modification such as member addition as well as initial key agreement.

In case that conference key agreement is performed on complete graph, it requires  $v(v-1)$  messages to be sent and one round message exchange, where  $v$  is the number of participants [1]. It is tried to construct efficient key agreement protocols by extending Diffie-Hellman [2] to groups in many researches. [1] is the first attempt to provide contributory key agreement. Here, they proposed a method performed on a logical ring network. It requires  $v-1$  rounds message exchange,  $v^2$  of traffic overhead and  $v^2/2$  of exponentiation. Another remarkable result is [3] in which it takes only 2 round message exchange, but it requires  $2v$  times broadcasting and  $n^2$  of exponentiation. [3] looks very efficient but it requires each entity to receive  $v-1$  messages in a single round. [4] is another contributory key agreement system which requires  $2v$  rounds message exchange,  $2v$  times unicast and 2 times broadcasting of traffic overhead and  $4v$  of exponentiation, but each entity needs still to receive  $v-1$  messages in each broadcasting round.

In this paper an efficient conference key agreement system for a static group is presented by employing symmetric balanced incomplete block design(SBIBD), one class of block designs. The protocol presented not only minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key.

Our protocol constructs a conference key which

takes the forms of  $\prod_{i=0}^{v-1} R_i$ , where  $v$  is the number of participants and  $R_i$  is a random number generated from member  $i$ . In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is  $O(v\sqrt{v})$ . Our protocol can be proved as computationally difficult to calculate as discrete logarithms.

The rest of this paper is organized as follows. Section 2 defines SBIBD. An algorithm for constructing  $(v, k + 1, 1)$ -configuration, one class of SBIBD, is described in section 3. Section 4 presents our conference key agreement protocol which is performed on a  $(v, k + 1, 1)$ -configuration and analyses the security and complexity issues. This paper concludes with summary and on-going and future work in section 5.

## 2 About $(v, k, \lambda)$ -configuration

Let  $V = \{0, 1, \dots, v - 1\}$  be a set of  $v$  elements. Let  $B = \{B_0, B_1, \dots, B_{b-1}\}$  be a set of  $b$  blocks, where  $B_i$  is a subset of  $V$  and  $|B_i| = k$ . For a finite incidence structure  $\sigma = \{V, B\}$ , if  $\sigma$  satisfies following conditions, then it is a balanced incomplete block design (BIBD) [5][6], which is called a  $(b, v, r, k, \lambda)$ -configuration.

1.  $B$  is a collection of  $b$   $k$ -subsets of  $V$  and this  $k$ -subsets are called the blocks.
2. Each element of  $V$  is related with exactly  $r$  of  $b$  blocks.
3. Every two objects of  $V$  appears simultaneously in exactly  $\lambda$  of  $b$  blocks.
4.  $k < v$ .

For a  $(b, v, r, k, \lambda)$ -configuration, if it satisfies  $k = r$  and  $b = v$ , then it is a symmetric balanced incomplete block design (SBIBD)[5][6] and it is called a  $(v, k, \lambda)$ -configuration. There are some relations among parameters  $b, v, r, k, \lambda$  that are necessary conditions for existence of a corresponding  $(b, v, r, k, \lambda)$ -configuration.

1. In a  $(b, v, r, k, \lambda)$ -configuration,  $bk = vr$  and  $r(k - 1) = \lambda(v - 1)$ .
2. For a  $(b, v, r, k, \lambda)$ -configuration,  $QQ^T = (r - \lambda)I + \lambda J$ ,  $I : v \times v$  identity matrix,  $J : v \times v$  matrix in which all the entities are 1's.
3. For a  $(b, v, r, k, \lambda)$ -configuration,  $b \geq v$ .

4. For a  $(v, k, \lambda)$ -configuration, every two blocks have exactly  $\lambda$  objects in common.

As shown above, it is not true that there exists a BIBD or SBIBD for arbitrary set of parameters  $b, v, r, k$  and  $\lambda$ . However there is no known sufficient condition on the existence of a certain  $(b, v, r, k, \lambda)$ -configuration or  $(v, k, \lambda)$ -configuration.

## 3 Design of an Algorithm to construct $(v, k + 1, 1)$ -configuration

Our key agreement system is based on the feature of  $(v, k + 1, 1)$ -configuration, that is, each block has  $k + 1$  elements and every two object appears simultaneously only one time in  $v$  blocks. In this section, we present an algorithm to generate an incidence structure  $\sigma = \{V, B\}$  satisfying the condition for a  $(v, k + 1, 1)$ -configuration in the case that  $k$  is a prime number and prove it. Notation used in this paper is as Table 1.

Table 1: Notation

notation	meaning
$X_i$	$i^{th}$ set in a family of set $X$
$X_{i,j}$	$j^{th}$ element of $X_i$
$X^i$	$i^{th}$ member of family of sets $X$ in which each member is a family of sets
$X^i_j$	$j^{th}$ set in a family of sets $X^i$
$\sigma\{V, X\}$	an incidence structure where $V$ is a set, $X$ is a family of sets and $X_i$ is a subset of $V$

We devised Algorithm 1 to generate  $(v, k + 1, 1)$ -configuration. Table 2 illustrates how to create  $Z = \{V, B\}$ ,  $V = \{0, 1, \dots, 12\}$ , in compliance with Algorithm 1. We now prove that this structure satisfies the conditions of a  $(v, k + 1, 1)$ -configuration.

*Definition 1.* On incidence structure  $Y$ , Sector  $S_i$  is the  $i^{th}$  family of  $k$  blocks,  $D_j \in S_i$ ,  $i = \lfloor j/k \rfloor$ .

For example, If  $k$  equals 3, then  $\lfloor 0/k \rfloor = \lfloor 1/k \rfloor = \lfloor 2/k \rfloor = 0$ . So,  $S_0 = \{D_0, D_1, D_2\}$ . There are  $k$  sectors in  $Y$ .

---

*Algorithm 1 :*  
Generating an  $(v, k + 1, 1)$ -configuration

---

*input :* prime number  $k$   
*output :*  $(v, k + 1, 1)$ -configuration

1. Generate a set  $V$ .  
 $v = k^2 + k + 1$   
 $V = \{0, 1, \dots, v - 1\}$
2. Construct two incidence structures  
 $X = \{V, C\}$  and  $Y = \{V, D\}$ .  
(a)  $C_{i,j}$ ,  $0 \leq i, j \leq k$ , has a value as following.

$$C_{i,j} = \begin{cases} 0 & \text{if } j = 0 \\ t & t = i \times k + j \quad \text{if } j \geq 1 \end{cases}$$

- (b)  $D_{i,j}$ ,  $0 \leq i \leq (k^2 - 1)$ ,  $0 \leq j \leq k$ , has a value as following.

$$D_{i,j} = \begin{cases} C_{0,t}, & t = [i/k] + 1 \\ & , \text{if } j = 0 \\ C_{j,t}, & t = (i + (j - 1) \times [i/k]) \bmod k + 1 \\ & , \text{if } j \geq 1 \end{cases}$$

3. Generate  $Z = \{V, B\}$  from  $X$  and  $Y$ .  
 $B_i \leftarrow C_i$   
 $B_{i+k+1} \leftarrow D_i$
- 

*Lemma 1.* For two elements  $D_{i_1, j_1}$  and  $D_{i_2, j_2}$ ,  
 $D_{i_1, j_1} \neq D_{i_2, j_2}$ , if  $j_1 \neq j_2$ .

*Proof.* From Algorithm 1-2-(a), if  $0 < j \leq k$ ,  $0 \leq i \leq k$  then  $C_{i,j} = i \times k + j$ . This means if  $j > 0$  then all the elements are distinct. And as shown in Algorithm 1-2-(b), an element of  $C_j$  is placed on  $j^{\text{th}}$  element of a certain block of  $Y$  if  $D_{i,j} = C_{j,t}$ ,  $t \neq 0$ .

*Lemma 2.* For a sector consisting of  $k$  blocks, the first element of each block has the same value and the other  $k^2$  elements are equal to  $V - C_0$ .

*Proof.* In the case that  $D_{i,0} = C_{0, [i/k] + 1}$ , the first element of  $k$  blocks on a sector have the same value. According to Algorithm 1-2-(b),  $D_{i,j} = C_{j,t}$ ,  $t = (i + (j - 1) [i/k]) \bmod k + 1$ . Since  $k$  is a prime number, each element except the first element of each block is distinct and these distinct  $k^2$  elements are equal to  $V - C_0$ .

*Lemma 3.* For incidence structure  $Y$ ,  $D_{a,j} =$

Table 2: A set of blocks on  $Z$  generated from Algorithm 1

X	Y
$C_0 = \{0, 1, 2, 3\}$	$D_0 = \{1, 4, 7, 10\}$
$C_1 = \{0, 4, 5, 6\}$	$D_1 = \{1, 5, 8, 11\}$
$C_2 = \{0, 7, 8, 9\}$	$D_2 = \{1, 6, 9, 12\}$
$C_3 = \{0, 10, 11, 12\}$	$D_3 = \{2, 4, 8, 12\}$
	$D_4 = \{2, 5, 9, 10\}$
	$D_5 = \{2, 6, 7, 11\}$
	$D_6 = \{3, 4, 9, 11\}$
	$D_7 = \{3, 5, 7, 12\}$
	$D_8 = \{3, 6, 8, 10\}$
Z	
$B_0 = \{0, 1, 2, 3\}$	$B_7 = \{2, 4, 8, 12\}$
$B_1 = \{0, 4, 5, 6\}$	$B_8 = \{2, 5, 9, 10\}$
$B_2 = \{0, 7, 8, 9\}$	$B_9 = \{2, 6, 7, 11\}$
$B_3 = \{0, 10, 11, 12\}$	$B_{10} = \{3, 4, 9, 11\}$
$B_4 = \{1, 4, 7, 10\}$	$B_{11} = \{3, 5, 7, 12\}$
$B_5 = \{1, 5, 8, 11\}$	$B_{12} = \{3, 6, 8, 10\}$
$B_6 = \{1, 6, 9, 12\}$	

$D_{b,j}$ ,  $j \geq 1$ , if  $b = ((a - c(j - 1)) \bmod k + k([a/k] + c)) \bmod k^2$ .

*Proof.* From Algorithm 1-2-(b),  $D_{a,j} = C_{j,t}$ . We now prove that  $D_{b,j} = C_{j,t}$ .  $t$  can be calculated from parameters  $b, j$  below. Then  $t$  obtained on this lemma is equal to that from Algorithm 1-2-(b). Therefore,  $D_{a,j} = D_{b,j}$ .

$$\begin{aligned} t &= (b + (j - 1) \times [b/k]) \bmod k + 1 \\ &= (((a - c(j - 1)) \bmod k + k([a/k] + c)) + (j - 1)[((a - c(j - 1)) \bmod k + k([a/k] + c))/k]) \bmod k + 1 \\ &= (((a - c(j - 1)) + (j - 1) \times ([a/k] + c)) \bmod k + 1 \\ &= (a + (j - 1)[a/k]) \bmod k + 1 \end{aligned}$$

Here, if  $D_{a,j}$  is in sector  $S_s$  then  $D_{b,j}$  is in  $S_{(s+c) \bmod k}$ . In case of  $c \equiv 0 \pmod k$ , then  $a = b$ .

*Lemma 4.* Each element of  $V$  appears in exactly  $k + 1$  times in  $Z$ .

*Proof.* According to Algorithm 1-2-(a),  $C_{i,0} = 0$ . Since  $0 \leq i \leq k$ , 0 appears  $k + 1$  times. The other  $v - 1$  elements,  $V - \{0\}$ , appear exactly once on  $X$ . From Lemma 3, each element of  $C_{0,j}$ ,  $1 \leq j \leq k$ , appears  $k$  times in a sector of  $Y$  and the rest  $k^2$  elements appear once in every sector of  $Y$ . Therefore, each element appears  $k + 1$  times in  $Z$ .

*Lemma 5.* Any pair of elements of  $V$  appears in exactly only once in  $Z$ .

*Proof.* The first element of  $V$  makes a pair with all the other elements and this pair appears once by designing rule of incidence structure(see Algorithm 1-2-(a)). Each elements of  $C_{0,j}, 1 \leq j \leq k$  makes a pair with  $V - C_0$  elements and it also appears once proven by Lemma 3. The rest  $k^2$  elements are now considered. For an arbitrary pair  $D_{a,j1} = D_{a,j2}, j1, j2 \geq 1$ , in order to make the same pair on other block  $D_b$ , the two elements should be on the same block. According to Lemma 4, if  $j1 = j2$ , then they are located on  $D_b$ . However, this case does not occur since  $j1 \neq j2$ . Therefore, any pair of elements of  $V$  appears in exactly only one time in  $Z$ .

*Theorem 1.*  $Z$  designed by Algorithm 1 satisfies the conditions of a  $(v, k + 1, 1)$ -configuration.

*Proof.*  $Z$  satisfied the conditions of the SBIBD by employing Lemma 4 and Lemma 5.

---

*Algorithm 2 :*

Generating an SBIBD in which block  $i$  contains element  $i$

---

*input* an incidence structure  $Z = \{V, B\}$  generated by Algorithm 1.

*output* generate  $Z' = \{V, E\}$ , where every block  $E_i$  includes object  $i$ .

$E_0 = C_0$  ;

for  $( i = 0 ; i < k^2 ; i ++ ) \{$

if  $( i \bmod k == 0 ) \{$

$Q = i/k ; B_{Q+1} = D_i ;$

$t = D_{i,Q+1} ; E_t = C_{Q+1} ; \}$

else  $\{$

$Q = \lceil i/k \rceil ; t = D_{i,Q} ;$

$E_t = D_i ; \}$

---

In this paper, we employ a  $(v, k, 1)$  – configuration for key agreement by mapping a block to a participant. This needs a  $(v, k, 1)$  – configuration in which each block  $i$

includes element  $i$ . Algorithm 2 is made for this reason. Table 3 shows the incidence structure  $Z' = \{V, E\}$  generated by Algorithm 2.

Table 3: rearranged blocks which block  $i$  contains element  $i$

$Z = \{V, B\}$	$Z' = \{V, E\}$
$C_0=B_0 = \{ \underline{0}, 1, 2, 3 \}$	$E_0 = \{ \underline{0}, 1, 2, 3 \}$
$C_1=B_1 = \{ 0, \underline{4}, 5, 6 \}$	$E_1 = \{ \underline{1}, 4, 7, 10 \}$
$C_2=B_2 = \{ 0, 7, \underline{8}, 9 \}$	$E_2 = \{ \underline{2}, 4, 8, 12 \}$
$C_3=B_3 = \{ 0, 10, \underline{11}, 12 \}$	$E_3 = \{ \underline{3}, 4, 9, 11 \}$
$D_0=B_4 = \{ \underline{1}, 4, 7, 10 \}$	$E_4 = \{ 0, \underline{4}, 5, 6 \}$
$D_1=B_5 = \{ 1, \underline{5}, 8, 11 \}$	$E_5 = \{ 1, \underline{5}, 8, 11 \}$
$D_2=B_6 = \{ 1, \underline{6}, 9, 12 \}$	$E_6 = \{ 1, \underline{6}, 9, 12 \}$
$D_3=B_7 = \{ \underline{2}, 4, 8, 12 \}$	$E_7 = \{ 2, 6, \underline{7}, 11 \}$
$D_4=B_8 = \{ 2, 5, \underline{9}, 10 \}$	$E_8 = \{ 0, 7, \underline{8}, 9 \}$
$D_5=B_9 = \{ 2, 6, \underline{7}, 11 \}$	$E_9 = \{ 2, 5, \underline{9}, 10 \}$
$D_6=B_{10} = \{ \underline{3}, 4, 9, 11 \}$	$E_{10} = \{ 3, 6, 8, \underline{10} \}$
$D_7=B_{11} = \{ \underline{3}, 5, 7, \underline{12} \}$	$E_{11} = \{ 0, 10, \underline{11}, 12 \}$
$D_8=B_{12} = \{ \underline{3}, 6, 8, \underline{10} \}$	$E_{12} = \{ 3, 5, 7, \underline{12} \}$

*Theorem 2.* In the incidence structure obtained from Algorithm 2, the  $i^{th}$  block contains element  $i$ .

*Proof.* It is clear that  $E_0$  includes element 0 because  $E_0 \leftarrow C_0$  and  $C_{0,0} = 0$ . In the case of  $0 \leq i < k^2$ , and  $i \bmod k = 0$ ,  $\{Q = i/k ; E_{Q+1} = D_i ; \}$  will be done. Here,  $E_1$  to  $E_k$  are assigned and each  $E_i$  contains member  $i$  because  $D_{i,0} = i/k + 1$  by Algorithm 1-2-(b). In the case of  $0 \leq i < k^2$ , and  $i \bmod k \neq 0$ ,  $\{ t = D_{i, \lceil i/k \rceil} ; E_t = D_i ; \}$  will be done. Blocks where have the same value of  $\lceil i/k \rceil$  are in the same sector according to Definition 1. Each sector from which  $1^{st}$  column is omitted has same elements  $V - C_0$  by Lemma 2. And every two column do not have common element by Lemma 1. So here, rest  $k^2 - k$  blocks assigned. Now,  $k$  blocks which are not assigned yet are  $D_{i, i/k+1}$ , where  $i \bmod k = 0$ . Each  $D_{i, i/k+1}$  can be found in  $C_{i/k+1}$  because  $D_{i, i/k+1} = C_{i/k+1, x}$ , where  $1 \leq x \leq k$  by Algorithm 1-2-(b).

Therefore each block  $i$  includes element  $i$  by Algorithm 2.

## 4 Design of a Conference Key Agreement System on $(v, k + 1, 1)$ -configuration

An efficient conference key agreement system is now constructed on  $(v, k + 1, 1)$ -configuration generated from Algorithm 2. In our protocol, every member contributes evenly to compute the same conference key,  $K = \prod_{n=0}^{v-1} R_n$ , which is computationally difficult to calculate as discrete logarithms, by 3 rounds message exchange and  $v\sqrt{v}$  traffic overhead, where  $v$  is the number of participants and  $R_n$  is a random number generated from member  $n$ . Algorithm 3 is the conference key agreement system we propose.

*Algorithm 3 :*

Construction of a Conference Key Agreement

*input :*  $N$  : a prime number  
 $g$  : a primitive element  
 $g \in \mathbb{Z}_N$ ,  
 $\mathbb{Z}_N = \{0, g, g^2, \dots, g^{N-1} = 1\}(\text{mod } N)$   
 $V$  : the set of participants  
 $X = \{V, C\}$  :  $(v, k+1, 1)$ -configuration generated from Algorithm 2  
*output :* each member computes the same key  $K$ .

1. Each member  $n$  on  $V$  defines two sets  $S1_n$  and  $S2_n$  as below.  
 $S1_n = \{x \mid n \in C_x \text{ and } n \neq x\}$   
 $S2_n = C_n - \{n\}$
2. Each member  $n$  generates a random number  $R_n$  and  $Q_n = g^{R_n}(\text{mod } N)$ .
3. Each member  $n$  sends  $Q_n$  to each member on  $S2_n$ .
4. Each member  $n$  generates  $M_i = \{g^{R_n}, R_n \times g^{R_n R_i}\}(\text{mod } N)$  and sends  $M_i$  to member  $i$ , where  $i \in S1_n$ .
5. Each member  $n$  computes  $K = R_n \prod R_i(\text{mod } N)$ , where  $i \in S2_n$ .
6. Each member  $n$  computes  $pK_{n,i} = g^{R_n R_i} \prod R_j(\text{mod } N)$ , where  $i \in S2_n$ ,  $j \in \{C_n - \{i\}\}$  and sends  $pK_{n,i}$  to member  $i$  on  $S2_n$ .
7. Each member computes the same conference key  $K = K \prod pK_{i,n}(\text{mod } N)$ , where  $i \in S1_n$ .

Table 4 indicates the procedure how member 7 on  $(13, 4, 1)$ -configuration like table 3 computes the conference key. The fact that every member computes the same key is proved in Theorem 2.

Table 4: Computation of the conference key on member 7 on  $X = \{V, C\}$

round	receiving messages or computation
define	$S1_7 = \{1, 8, 12\}$ $S2_7 = \{2, 6, 11\}$
1 <sup>st</sup>	$g^{R_1}, g^{R_8}, g^{R_{12}}$
2 <sup>nd</sup>	$\{g^{R_2}, R_2 \times g^{R_2 R_7}\},$ $\{g^{R_6}, R_6 \times g^{R_6 R_7}\},$ $\{g^{R_{11}}, R_{11} \times g^{R_{11} R_7}\}$
compute	$x = g^{R_2 R_7}; R_2 = (R_2 \times g^{R_2 R_7})/x$ $x = g^{R_6 R_7}; R_6 = (R_6 \times g^{R_6 R_7})/x$ $x = g^{R_{11} R_7}; R_{11} = (R_{11} \times g^{R_{11} R_7})/x$ $K = R_7 \times R_2 \times R_6 \times R_{11}$
3 <sup>rd</sup>	$pK_{1,7}, pK_{8,7}, pK_{12,7}$
compute	$K = K \times (pK_{1,7}/g^{R_1 R_7}) \times$ $(pK_{8,7}/g^{R_8 R_7}) \times (pK_{12,7}/g^{R_{12} R_7})$ because $pK_{1,7} = g^{R_1 R_7} \times R_1 \times R_4 \times R_{10},$ $pK_{8,7} = g^{R_8 R_7} \times R_0 \times R_8 \times R_9 \text{ and}$ $pK_{12,7} = g^{R_{12} R_7} \times R_3 \times R_5 \times R_{12}$

*Theorem 3.* According to Algorithm 3, every member of  $V$  computes the same conference key  $K = \prod R_n$ , where  $n \in V$ .

*Proof.* Let an arbitrary member  $n$  define  $S1_n = \{a_1, a_2, \dots, a_k\}$  and  $S2_n = \{b_1, b_2, \dots, b_k\}$ . Cardinality of the set  $S1_n$  and  $S2_n$  is  $k$  because every block contains  $(k + 1)$  elements and every element appears on  $(k + 1)$  blocks in a  $(v, k + 1, 1)$ -configuration.

For an element  $a$  on  $S1_n$ , it is true that if  $a$  is a member of  $S1_n$  then  $n$  is a member of  $S2_a$ . Similarly, for an element  $b$  on  $S2_n$ , if  $b$  is a member of  $S2_n$  then  $n$  is a member of  $S1_b$ . This means member  $n$  receives messages from members on  $S1_n$  while it sends messages to members on  $S2_n$  and receives from members on  $S2_n$  while sending to members on  $S2_n$ .

Member  $n$  can compute  $\prod R_b$ , multiplication of  $(k + 1)$  random numbers, from the messages received in the 2<sup>nd</sup> round, where  $b \in C_n$ . The message  $pK_{a,n}$ , arrived in the 3<sup>rd</sup> round, consists of  $k$  random numbers generated from members on  $C_a - \{n\}$ , where  $a \in S1_n$ . So mem-

ber  $n$  come to gain multiplication of  $k^2$  random numbers and each of these  $k^2$  random numbers has individual source because every two block has only one common element in a  $(v, k + 1, 1)$ -configuration and this element is  $n$ . Therefore arbitrary member  $n$  computes the same conference key  $K = \prod R_n$ , where  $n \in V$ .

*Theorem 4.* The key computed from Algorithm 3 is computationally difficult to calculate as discrete logarithms.

*Proof.* Three rounds message exchanging is performed in algorithm 3 for key agreement. In the first round, while a member  $n$  is sending  $g^{R_n}$  to member  $j$ , he/she receives  $g^{R_i}$  from member  $i$ . In the second round, member  $n$  receives  $g^{R_j}$  and  $Y = R_j \times g^{R_n R_j}$ . It is possible for the member  $n$  to obtain  $R_j$  from  $Y$  because of  $R_j = Y / (g^{R_j})^{R_n}$ . But this critical information  $R_j$  can be protected because the thing provided to eavesdropper is only  $g^{R_j}$  and  $g^{R_n}$ . In the third round, member  $n$  receives  $Y = M \times g^{R_n R_i}$ , message  $M$  can be calculated by only member  $n$  in the same manner. Therefore finding the key generated from this algorithm is a discrete logarithm problem.

## 5 Conclusion

An efficient key agreement system is presented for group communication. Our protocol minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key and it is computationally difficult to calculate as discrete logarithms for an eavesdropper to find the key.

Proposed protocol constructs a conference key which takes the forms of  $\prod_{i=0}^{v-1} R_i$ , where  $v$  is the number of participants and  $R_i$  is a random number generated from member  $i$ . In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is  $O(v\sqrt{v})$ , where every member sends and receives  $k$  messages in each round equally.

This algorithm is well performed when the number of participants is  $v = k^2 + k + 1$ . We are studying the method to apply our protocol in the case of arbitrary number of participants.

## References

- [1] I.Ingemarrson, D.T.Tang, C.K.Wong, A conference key distribution System, IEEE Trans. Inform. Theory vol.28, pp.714-720, 1982.
- [2] Whit Diffie and Martin Hellman, New Direction in cryptography, IEEE Trans Inform. Theory, vol.22, no.6, pp644-654, 1976.
- [3] Burmester, Y.Desmedt, "A Secure and Efficient Conference Key Distribution System, LNCS vol.950, pp.275-286, 1994.
- [4] M Steiner, G.Tsudik and M.Waidner, Diffie-Hellman Key Distribution Extended to Groups, ACM CCS96, pp31-37, 1996.
- [5] M.K.Bennett Affine and projective geometry Wiley & Sons, 1995.
- [6] C.L.Liu, Introduction to Combinatorial Mathamatics, McGraw-Hill,NY, pp.359-383, 1968.