Infinite classes of conference key distribution systems constructed from difference families

S. Georgiou, C. Koukouvinos, and E. Lappas
Department of Mathematics
National Technical University of Athens
Zografou 15773, Athens, Greece

Abstract

Combinatorial designs is a powerful tool because of their beautiful combinatorial structure that can help in many applications, such as coding theory or cryptography. A conference key distribution system is a scheme to design a conference key, and then to distribute this key to only participants attending the conference in order to communicate with each other securely. In this paper we present an efficient conference key distribution system using difference families. Using techniques for creating the conference key and for performing authentication based on identification information, the communication protocol is designed. Applying the known results on difference families we obtain many new infinite classes of conference key distribution systems. In special classes of difference families the message overhead is $O(v\sqrt{tv})$, where v is the number of participants and t is the number of the k-elements subsets that consist the difference family. The security of the presented protocol, which is an important problem in the construction of a secure system, is proved to be as computationally difficult to calculate as factoring and discrete logarithms.

Key words and phrases: Difference family, incidence matrix, conference key distribution system, algorithm.

1 Introduction

A cryptosystem can help users to establish secure communication channel in open environment. Diffie and Hellman [5] introduced the concept of public key cryptography, which is referred to as a key distribution system

(KDS). A conference key distribution system (CKDS) [2] is a scheme to generate a common secret key called *conference key*, and to distribute this key to all participants attending the conference in order to communicate with each other securely.

Authentication [14] is the most important of the security services, because all other security services depend upon it. It is the mean of gaining confidence that people or things are who or what they claim to be. An important CKDS considering authentication was proposed by Shamir in [16], where he utilizes an ID-based public key system. User's public key contains user's name and address. Fiat and Shamir [6] suggested an authentication mechanism employing discrete logarithm. Okamoto [12] proposed an identity-based key distribution system. Ingemarsson, Tang and Wang [7] presented a CKDS on a ring topology network. Koyama and Ohta [10] proposed identity-based CKDS (ICKDS) on ring, compete graph and star topology networks. Shimbo and Kawamura [17] analyzed several CKDSs.

In the case that an ICKDS is performed, a common conference key should be created in order for all participants of the conference to communicate mutually. Assume that each user (participant) has his own key and the common conference key should be created using these keys. One possible way to generate the public key is by requiring each user to send their own key to every other user. Then the computation of the public key can be perform at each users cite. This method requires $v \times (v-1)$ messages [7] to be sent, where v is the number of participants to the conference, and one round of message exchange. The public conference key is computed in each users site as $r_1 \times r_2 \cdots \times r_v$, where r_i is user i's secret key. The message overhead requires $O(v^2)$ and that will cause the conference to be delayed when v gets to be large. Recently, a method which develops a conference key distribution system based on $SBIB$ designs was given in [3].

Designs and cryptography seems to be closely connected. Many designs have been used for the construction of many cryptosystems (see for example [3, 13]). In this paper, we present an efficient conference key distribution system which is a generalization of the key given in [3]. To achieve this, a difference family (see [8]) is applied for generating the conference key and to distribute this to participants. Through this technique, for creating a conferences key and for performing mutual authentication based on identification information, the communication protocol is designed.

The protocol presented minimizes the message overhead for generating a conference key. In some special classes of difference families the overhead is $O(v\sqrt{tv})$, where v is the number of participants attending the conference and t is the number of subsets that compose the difference family. This protocol needs two rounds of message exchange. The security of the mechanism, which is an important problem in the construction of a secure system, can be proved to be as computationally difficult to calculate as factoring and discrete logarithms.

In the next section we present some preliminary results and basic definitions concerning difference families.

2 Difference families

The main purpose of this section is to provide the basic definitions and notations we shall need to develop our method. For the following definitions we refer to [8].

Definition 1 A (v, k, λ) *difference set* $D = \{d_1, d_2, \ldots, d_k\}$ is a collection of k residues modulo v, such that for any residue $a \not\equiv 0$ $(mod\ v)$ the congruence

$$d_i - d_j \equiv a\ (mod\ v)$$

has exactly λ solution pairs (d_i, d_j) with d_i and d_j in D. □

Example 1 Some non-trivial difference sets are:

$$D_1 = \{1, 2, 4\}\ (mod\ 7), \quad (v, k, \lambda) = (7, 3, 1)$$

$$D_2 = \{0, 3, 5, 6\}\ (mod\ 7), \quad (v, k, \lambda) = (7, 4, 1)$$

$$D_3 = \{0, 1, 3, 9\}\ (mod\ 13), \quad (v, k, \lambda) = (13, 4, 1)$$

$$D_4 = \{1, 4, 5, 6, 7, 9, 11, 16, 17\}\ (mod\ 19), \quad (v, k, \lambda) = (19, 9, 4)$$ □

Theorem 1 ([8]) *There exist difference sets with parameters* $(v, k, \lambda) = (n^2 + n + 1, n + 1, 1)$, *when* n *is prime power. Difference sets with these parameters are called* projective planes. □

Theorem 2 ([8]) *There exist difference sets with parameters* $(v, k, \lambda) = 4t - 1, 2t - 1, t - 1$ *for all orders* $4n$ *for which a Hadamard matrix of order* $4n$ *exist.* □

For the existence of Hadamard matrices see [4, 15]. Many classes of difference sets are known. For more details the interesting reader, should consult [1, 9].

Difference sets have a rich combinatorial structure but they cannot exist for all possible values (v, k, λ). A generalization of difference sets, that can exist in many more cases, is difference families. For the following definition and more results, see [8].

Definition 2 Let G be an abelian group of order v. Then t subsets of G, $B_i = \{b_{i,1}, b_{i,2}, \ldots, b_{i,k}\}$ consist of k elements each, $(1 \le i \le t)$ form a $(v, k, \lambda; t)$ *difference family* (or *difference system*) if every nonzero element of G occurs λ times among the differences $b_{i,x} - b_{i,y}$ $(i = 1, 2, \ldots, t;\ x, y = 1, 2, \ldots, k)$. The sets B_i are called *base blocks*. In many cases, where t is obvious or can be easily calculated, we use notation (v, k, λ) to denote the difference family. □

Remark 1 It is obvious that if $t = 1$, then B_1 is an abelian (v, k, λ) difference set and thus difference families is a generalization of difference sets. □

Example 2 The base blocks

$$B_1 = \{0, 1, 2, 4, 8\},\quad B_2 = \{0, 1, 3, 6, 12\},\quad B_3 = \{0, 2, 5, 6, 9\}$$

form a difference family with parameters $t = 3$ and $(v, k, \lambda) = (13, 5, 5)$. □

Definition 3 An *incidence matrix* A_i of the base block B_i is the circulat matrix with first row the sequence s_i of length v with entries from $\{0, 1\}$ defined by $s_{i,j} = \begin{cases} 1, & j \in B_i \\ 0, & j \notin B_i \end{cases}$, where $j = 0, 1, \ldots, v-1$ and $i = 1, 2, \ldots, t$. □

Example 3 The incidence matrix of base block B_1, as this is given in Example 2 is given in Table 1. Observe that in the first row we have the ones in positions $j \in B_1$ and zero elsewhere. Similarly, one can obtain the incidence matrices of base blocks B_2 and B_3. □

Theorem 3 ([8]) *If there exist a difference family with t subsets on an additive group G and with parameters (v, k, λ) then the equation*

$$\lambda(v - 1) = tk(k - 1) \tag{1}$$

holds. □

Condition (1) of Theorem 3 is necessary but not sufficient for the existence of difference family with parameters (v, k, λ). For instance, it was proved in [11] that there is no difference family with parameters $(v, k, \lambda) = (111, 11, 1)$ and $t = 1$ even thought the condition (1) of Theorem 3 is satisfied for $t = 1$.

There are many classes of difference families, some of which are given in the next Theorems. For more details see [8].

Table 1: The incidence matrix of base block B_1 of Example 2.

$A_1 =$

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	1	1	0	1	0	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0	0	0	1	0	0	0
2	0	0	1	1	1	0	1	0	0	0	1	0	0
3	0	0	0	1	1	1	0	1	0	0	0	1	0
4	0	0	0	0	1	1	1	0	1	0	0	0	1
5	1	0	0	0	0	1	1	1	0	1	0	0	0
6	0	1	0	0	0	0	1	1	1	0	1	0	0
7	0	0	1	0	0	0	0	1	1	1	0	1	0
8	0	0	0	1	0	0	0	0	1	1	1	0	1
9	1	0	0	0	1	0	0	0	0	1	1	1	0
10	0	1	0	0	0	1	0	0	0	0	1	1	1
11	1	0	1	0	0	0	1	0	0	0	0	1	1
12	1	1	0	1	0	0	0	1	0	0	0	0	1

Theorem 4 ([8]) *There exist $(v, k, \lambda; t)$ difference families for the following values:*

a. $(v, 3, 1)$, $v \equiv 1 \ (mod\ 6)$.

b. $(v, 3, 2)$, $v =$ *16, 28, 40.*

c. $(v, 3, 3)$, $v = 2t + 1$, $t \geq 1$.

d. $(v, 4, 1)$, $v = 12t + 1 < 10^6$ *when v is prime and $v < 10^6$, or v is prime power that is not a prime or t=1, 3-11, 13-16, 18-21, 23, 25, 26, 28-31, 33-36, 38-41, 43, 45, 46, 48, 50.*

e. $(v, 5, 1)$,$v = 20t + 1 < 10^4$ *when v is a prime power and $v < 10^4$, or t=1-3, 5-10, 12-15, 18, 20-28, 30, 32-35, 38-44, 47-50.* □

The following Theorem is given in [18].

Theorem 5 *Suppose v is a prime power, and $\lambda(v-1) \equiv 0 \ (mod\ k(k-1))$. Then a (v, k, λ) difference family over F_v exists if one of the following holds:*

1. λ *is a multiple of* $k/2$ *or* $(k-1)/2$.

2. $\lambda \geq k(k-1)$.

3. $v > \binom{k}{2}^{k(k-1)}$ □

For more results and details on difference families the reader should consult [1, 8].

3 The design of a conference key distribution system based on difference families

In order the v participants to communicate mutually, the conference key should be created using all of their personal private keys. The minimal message transmission overhead for this process must be guaranteed. In this paper, we shall use the incidence matrices of the base blocks of the difference family to obtain the desirable conference key. Row i and column j of these incidence matrices correspond to participant i and key j, respectively.

We present the method to obtain a conference key from a difference family as this is described in Theorem 6.

The method:

Step 0a. Select a $(v, k, \lambda; t)$ difference family, where v is the number of people attending the conference.

Step 0b. Create the incidences matrices $A_s = (a_{s;i,j})$ of the base blocks B_s for all $s = 1, 2, \dots, t$.

Step 1a. User i receives key r_j from user j iff $j \in \bigcup_s \{j : a_{s;i,j} = 1\} \setminus \{i\}$.

Step 1b. User i calculate the products $k_{s;i,j}$, where $k_{s;i,j}$ is the product of r_m's, $m \in \{j : a_{s;i,j} = 1\} \setminus \{j\}$, for all $s = 1, 2, \dots, t$.

Step 2a. User i receive products $k_{s,j,i}$ from user j iff $j \in \bigcup_s \{j : a_{s;j,i} = 1\} \setminus \{i\}$.

Step 2b. Then, user i can calculate, from his site, the conference key as $K = r_i^\lambda \prod_{s=1}^{t} \prod_{a_{s;i,j}=1} k_{s;j,i}$.

Now we are in position to prove Theorem 6. This theorem explains how and why this key is calculated this way, as it is shown in Step 2b of our method.

Theorem 6 *The conference key based, on the $(v, k, \lambda; t)$ difference family, is calculated by user i using the equation*

$$K = r_i^\lambda \prod_{s=1}^{t} \prod_{a_{s;i,j}=1} k_{s;j,i}, \tag{2}$$

where $A_s = (a_{s;i,j})$ *are the incidence matrices of the base blocks* B_s, $s = 1, 2, \ldots, t$ *and* $k_{s;j,i}$, $s = 1, 2, \ldots, t$ *are product of keys calculated as it is shown in step 1b of our method.*

Proof. From the definition of a $(v, k, \lambda; t)$ difference family we conclude that each row and column of the $v \times v$ incidence matrices is consist from exactly $v - k$ zeros and k ones. In order to achieve the desirable result, all users to communicated mutually, the common conference key should be generated using all $r_0, r_1, \ldots, r_{v-1}$ user's personal secret keys. To create this key we use the selected difference family and the following two steps. On the first step, user i receives $t(k-1)$ keys and compute tk product, each of which is composed by $(k-1)$ distinguish keys. The fact that the keys are all different arise from the structure of difference families, see method (step 1a and 1b). On the second step, user i receives $t(k-1)$ products consisting of $(k-1)$ keys each. Thus the number of keys in the collection of product he posses (including his own products he needs) is $tk(k-1)$. Applying equation (1) in the parameters of this difference family we conclude that $tk(k-1)$ (or $(v-1)$) personal secrete keys appear λ times in this collection of products. So we can obtain the desirable common key by multiply all products together with λ times user's i personal secret key r_i. Thus the final public common conference key can be calculated by user i as $K = r_i^{\lambda} \prod_{s=1}^{t} \prod_{a_{s;i,j}=1} k_{s;j,i}$. □

For example, thirteen participants take part in a conference and each of them has his own secret key. Each participant computes a conference key based on a $(v, k, \lambda) = (13, 3, 1)$ difference family which is consist from $t = \frac{\lambda(v-1)}{k(k-1)} = 2$ subsets B_1, B_2 of Z_{13} with $k = 3$ elements each. The two incidence matrices of base blocks $B_1 = \{0, 1, 4\}$ and $B_2 = \{0, 2, 7\}$ are given in Table 2.

In order to generate a conference key, each user receive some keys from other users chosen by employing the structure of these matrices. Two steps are required to calculate the public conferences key. User i receives key r_j from user j in the case $a_{1;i,j} = 1$ or $a_{2;i,j} = 1$. We know describe this process from the viewpoint of user 0. First user 0 receives keys r_1, r_4 (because $a_{1;0,1} = 1$ and $a_{1;0,4} = 1$) and keys r_2, r_7 (because $a_{2;0,2} = 1$ and $a_{2;0,7} = 1$). Then he calculate $k_{1;0,0} = r_1 r_4$, $k_{1;0,1} = r_0 r_4$, $k_{1;0,4} = r_1 r_0$, $k_{2;0,0} = r_2 r_7$, $k_{2;0,2} = r_0 r_7$, $k_{2;0,7} = r_2 r_0$, where $k_{s;0,j}$ is the product of r_b's, $b \in \{\ell : a_{s;0,\ell} = 1\} - \{j\}$. Simultaneously, all other users to the same process. Next, user i receives $k_{s;j,0}$ if $a_{s;j,0} = 1$, $s = 1, 2$. Thus user 0 receives $k_{1;9,0}$, $k_{1;12,0}$, $k_{2;6,0}$, $k_{2;11,0}$ from users 6, 9, 11, 12. Then the conference key K is calculated from the relation: $K = r_0 k_{1;0,0} k_{1;9,0} k_{1;12,0} k_{2;0,0} k_{2;6,0} k_{2;11,0}$.

A summary for the two steps needed in our example (using (13,3,1)

Table 2: The two incidence matrices of an $(13, 3, 1)$ difference family

$$A_1 = (a_{1;i,j}) = \begin{pmatrix}
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$A_2 = (a_{2;i,j}) = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

difference family) are given in Table 3.

We had constructed a conference key distribution system to be used in the communication of the users. To do so, we generate the common key based on the keys we receive from all users. We cannot guarantee whether each of these keys are correct and have been sent from the right person. Sometimes things are not what they seems to be. In the next section we utilize user's identity information for authentication in order to solve this problem.

Table 3: Conference key constructed using a difference family $(13, 3, 1)$

user	Step 1 (Define)	Step 2 (Multiply)
0	$k_{1;0,0} = r_1 r_4$, $k_{1;0,1} = r_0 r_4$, $k_{1;0,4} = r_0 r_1$, $k_{2;0,0} = r_2 r_7$, $k_{2;0,2} = r_0 r_7$, $k_{2;0,7} = r_0 r_2$	$r_0 k_{1;0,0} k_{1;9,0} k_{1;12,0}$ $k_{2;0,0} k_{2;6,0} k_{2;11,0}$
1	$k_{1;1,1} = r_2 r_5$, $k_{1;1,2} = r_1 r_5$, $k_{1;1,5} = r_1 r_2$, $k_{2;1,1} = r_3 r_8$, $k_{2;1,3} = r_1 r_8$, $k_{2;1,8} = r_1 r_3$	$r_1 k_{1;0,1} k_{1;1,1} k_{1;10,1}$ $k_{2;1,1} k_{2;7,1} k_{2;12,1}$
2	$k_{1;2,2} = r_3 r_6$, $k_{1;2,3} = r_2 r_6$, $k_{1;2,6} = r_2 r_3$, $k_{2;2,2} = r_4 r_9$, $k_{2;2,4} = r_2 r_9$, $k_{2;2,9} = r_2 r_4$	$r_2 k_{1;1,2} k_{1;2,2} k_{1;11,2}$ $k_{2;0,2} k_{2;2,2} k_{2;8,2}$
3	$k_{1;3,3} = r_4 r_7$, $k_{1;3,4} = r_3 r_7$, $k_{1;3,7} = r_3 r_4$, $k_{2;3,3} = r_5 r_{10}$, $k_{2;3,5} = r_5 r_{10}$, $k_{2;3,10} = r_3 r_5$	$r_3 k_{1;2,3} k_{1;3,3} k_{1;12,3}$ $k_{2;1,3} k_{2;3,3} k_{2;9,3}$
4	$k_{1;4,4} = r_5 r_8$, $k_{1;4,5} = r_4 r_8$, $k_{1;4,8} = r_4 r_5$, $k_{2;4,4} = r_6 r_{11}$, $k_{2;4,6} = r_4 r_{11}$, $k_{2;4,11} = r_4 r_6$	$r_4 k_{1;0,4} k_{1;3,4} k_{1;4,4}$ $k_{2;2,4} k_{2;4,4} k_{2;10,4}$
5	$k_{1;5,5} = r_6 r_9$, $k_{1;5,6} = r_5 r_9$, $k_{1;5,9} = r_5 r_6$, $k_{2;5,5} = r_7 r_{12}$, $k_{2;5,7} = r_5 r_{12}$, $k_{2;5,12} = r_5 r_7$	$r_5 k_{1;1,5} k_{1;4,5} k_{1;5,5}$ $k_{2;3,5} k_{2;5,5} k_{2;11,5}$
6	$k_{1;6,6} = r_7 r_{10}$, $k_{1;6,7} = r_6 r_{10}$, $k_{1;6,10} = r_6 r_7$, $k_{2;6,6} = r_8 r_0$, $k_{2;6,8} = r_6 r_0$, $k_{2;6,0} = r_6 r_8$	$r_6 k_{1;2,6} k_{1;5,6} k_{1;6,6}$ $k_{2;4,6} k_{2;6,6} k_{2;12,6}$
7	$k_{1;7,7} = r_8 r_{11}$, $k_{1;7,8} = r_7 r_{11}$, $k_{1;7,11} = r_7 r_8$, $k_{2;7,7} = r_9 r_1$, $k_{2;7,9} = r_7 r_1$, $k_{2;7,1} = r_7 r_9$	$r_7 k_{1;3,7} k_{1;6,7} k_{1;7,7}$ $k_{2;0,7} k_{2;5,7} k_{2;7,7}$
8	$k_{1;8,8} = r_9 r_{12}$, $k_{1;8,9} = r_8 r_{12}$, $k_{1;8,12} = r_8 r_9$, $k_{2;8,8} = r_{10} r_2$, $k_{2;8,10} = r_8 r_2$, $k_{2;8,2} = r_8 r_{10}$	$r_8 k_{1;4,8} k_{1;7,8} k_{1;8,8}$ $k_{2;1,8} k_{2;6,8} k_{2;8,8}$
9	$k_{1;9,0} = r_9 r_{10}$, $k_{1;9,9} = r_0 r_{10}$, $k_{1;9,10} = r_0 r_9$, $k_{2;9,9} = r_{11} r_3$, $k_{2;9,11} = r_9 r_3$, $k_{2;9,3} = r_9 r_{11}$	$r_9 k_{1;5,9} k_{1;8,9} k_{1;9,9}$ $k_{2;2,9} k_{2;7,9} k_{2;9,9}$
10	$k_{1;10,1} = r_{10} r_{11}$, $k_{2;10,12} = r_{10} r_4$, $k_{1;10,11} = r_1 r_{10}$, $k_{2;10,10} = r_{12} r_4$, $k_{1;10,10} = r_1 r_{11}$, $k_{2;10,4} = r_{10} r_{12}$	$r_{10} k_{1;6,10} k_{1;9,10} k_{1;10,10}$ $k_{2;3,10} k_{2;8,10} k_{2;10,10}$
11	$k_{1;11,2} = r_{11} r_{12}$, $k_{2;11,0} = r_{11} r_5$, $k_{1;11,12} = r_2 r_{11}$, $k_{2;11,11} = r_0 r_5$, $k_{1;11,11} = r_2 r_{12}$, $k_{2;11,5} = r_{11} r_0$	$r_{11} k_{1;7,11} k_{1;10,11} k_{1;11,11}$ $k_{2;4,11} k_{2;9,11} k_{2;11,11}$
12	$k_{1;12,3} = r_{13} r_0$, $k_{2;12,1} = r_{12} r_6$, $k_{1;12,0} = r_3 r_{12}$, $k_{2;12,12} = r_1 r_6$, $k_{1;12,12} = r_3 r_0$, $k_{2;12,6} = r_{12} r_1$	$r_{12} k_{1;8,12} k_{1;11,12} k_{1;12,12}$ $k_{2;5,12} k_{2;10,12} k_{2;12,12}$

4 The design of a conference key distribution system for authentication

One of the most important problems in cryptography is to create a system to provide authentication services. The following steps can be applied to create the secret information needed for the secure authentication in the network.

1. Choose p, q large primes (approximately 100 digits each) and compute their product $n = pq$.

2. Select a relative large integer e which is satisfy $gcd(e, (p-1)(q-1)) = 1$, where $gcd(a, b)$ denotes the greatest common divisor of a and b. Solve the equivalence $ed \equiv 1 \ mod \ (p-1)(q-1)$ to find the integer d.

3. Obtain an integer g such that $g \in GF(p) \bigcap GF(q)$.

4. Using user's i information ID and the integer d, found in 2, compute the secret information S_i.

A system shall distributes (e_i, g, n) to all users and user i keeps (d, S_i) secret. In order to describe the procedure and present the protocol to use, we use the following notation:

Notation 1 *When we write $(i \rightarrow j : M)$ we wish to denote that user i transmit the information M to user j.*

Notation 2 *The symbolism $(i :)$ indicated that user i stays at his site and does verification or computation.*

Using the above and a $(v, k, \lambda; t)$ difference family we can now present a protocol that achieves user authentication.

The protocol:

1. $i \rightarrow j : (ID_i, (X_i)^e_j, Y_i, t_i)$, $X_i = g^{er_i} \ (mod \ n)$, $Y_i = S_i g^{C_{i_1} r_i} \ (mod \ n)$, where $C_{i_1} = h(X_i, t_i)$ and $j \in \{j : a_{s;i,j} = 1, s = 1, 2, \ldots, t\}$.

   In other words, user i creates two information X_i and Y_i for authentication, encrypts X_i using e_j and transmits $(ID_i, (X_i)^{e_j}, Y_i, t_i)$ to user j, where r_i is a secret key of user i and h is a common hashing function which it is known to all users.

2. $j : X_i = ((X_i)^{e_j})^{d_j}$, $ID_i = Y_i^e / X_i^{C_{i_2}}$, where $C_{i_2} = h(X_i, t_i)$.

   This means that user j decrypts $(X_i)^{e_j}$ using d_j to obtain X_i. Since the hashing function h is common, user j apply this function to the received, from user i, information to authenticate user i. If $ID_i = Y_i^e / Y_i^{C_{i_2}}$, then the claim is acceptable.

3. $j \rightarrow p : (ID_j, (X_{j_p})^{e_p}, Y_{j_p}, t_j)$, $X_{j_p} = X_{p_1} X_{p_2} \ldots X_{p_{t(k-1)}}$, where $p_i \in \bigcup_{s=1}^{t} \{i : a_{s;j,i}\} \setminus \{p\}$, $Y_{j_p} = S_j g^{C_{j_1} r_j} \ (mod \ n)$, where $C_{j_1} = h(X_{j_p}, t_j)$.

User j receives $t(k-1)$ keys transmitted from users i belonging to the set $\bigcup_{s=1}^{t}\{i : a_{s;j,i} = 1\}$. Then he computes X_{j_p} and Y_{j_p}, and sends $(ID_j, (X_{j_p})^{e_p}, t_j)$ to user p.

4. $p : X_{j_p} = ((X_{j_p})^{e_p})^{d_p}$, $ID_j = Y_{j_p}^e / X_{j_p}^{C_{j_2}}$, where $C_{j_2} = h(X_{j_p}, t_j)$.
   In this way X_{j_p} can be computed when $(X_{j_p})^{e_p}$ was decrypted using d_p. Then user p authenticates user's j identity using the information obtained from this user. When $ID_j = Y_{j_p}^e / X_{j_p}^{C_{j_2}}$ then the authentication process is successful and the user is who he claims to be.

Theorem 7 *If $ID_i = Y^e / X_i^{C_{i_1}}$, then user j gains confidence that the information transmitted from user i, for the generation of a conference key, is correct.*

Proof. We have that $Y_i^e / X_i^{C_{i_1}} = (S_i g^{C_{i_1} r_i})^e / (g^{e r_i})^{C_{i_2}} = S_i^e$, if $C_{i_1} = C_{i_2}$. Since $S_i = ID_i^d$, we have that $(ID_i^d)^e$ is ID_i from Euler's theorem. □

In order to compute a conference key, user p utilize his own secret key and all X_{j_p}, $p \in \{j : a_{s;p,j} = 1\}$. Since each secret key and e appear λ times and $\lambda(v-1)$ times in X_{j_p}'s, respectively, user p calculates a conference key using the equation

$$K = X_{j_{p_1}} X_{j_{p_2}} \cdots X_{j_{p_{t(k-1)}}} g^{e_p^\lambda r_p^\lambda}.$$

From the equation above we can easily conclude that, as smaller λ is, the better computational efficiency and less complexity we have in the above representation of the key. Thus for small λ the conference key will be calculated faster and thus the conference will not be delayed.

5 Security and complexity of the key

In this section we analyze the proposed conferences key distribution system which is developed using a $(v, k, \lambda; t)$ difference family.

The first and second step requires $vt(k-1)$ messages to be transmitted when each of these is applied. Using expression (1) given in Theorem 6 we can easily conclude that the complexity is $vt(k-1)$. From $\lambda(v-1) = tk(k-1)$, k is determined by the values of v, t and λ. So in the case that λ and t get smaller the complexity gets smaller as well. Since the smallest possible value of λ is 1 and the smallest possible value of t is 1 (this is the case of $(v, k, 1)$ difference set, see Theorem 1) the smallest possible value of complexity is $O(v\sqrt{v})$. When λ is 1, many infinite classes of difference families exist (for example Theorem 4), then the complexity is $O(v\sqrt{tv})$ and when we have a random difference family with parameters $(v, k, \lambda; t)$ then the complexity is, as it is shown previously, $vt(k-1)$. This complexity is better than the complexity $O(v^2)$, of the classical conference key constructions (all user submit their keys to each other), in all cases for which $O(tk) < O(v)$.

From the existing infinite classes of $(v, k, 1; 1)$ difference sets (projective planes) we construct infinite families of conference key distribution systems which have complexity $O(v\sqrt{v})$. Also from the the existing infinite classes of $(v, k, 1; t)$ difference families we construct infinite classes of conference key distribution systems which have complexity $O(v\sqrt{tv})$. From all other known infinite classes of $(v, k, \lambda; t)$ difference families we construct infinite classes of conference key distribution systems which have complexity $O(vt(k-1))$.

So, as we have shown this protocol can be computed easy enough $(O(v\sqrt{v}))$ but is it secure? The security of the protocol is a significant problem in the construction of a secure system. In this protocol there is no way to reveal secret information S_i, given e, n and d, because no polynomial algorithms that solve the factorization problem are known. To secure this mechanism we will keep the secret keys r_i well protected. Moreover, for X_i given, it is very difficult to calculate r_i because discrete logarithms need to be found, and that problem is generally a very hard problem. Therefore, security of the communication protocol is computationally ensured.

References

[1] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, London, 1986.

[2] C. Chang, T. Wu, and C. Chen, The design of conference key distribution system, *Proc. of ASIACRYPT'92*, (1992), 11.1–11.6.

[3] I. Chung, W. Choi, Y. Kim, and M. Lee, The design of conference key distribution system employing a symmetric balanced incomplete block designs, *Information Processing Letters*, 81 (2002), 313–318.

[4] R. Craigen, Hadamard Matrices, *The CRC Handbook of Combinatorial Designs*, eds. C. J. Colbourn and J. H. Dinitz, CRC Press, Boca Raton, Fla., (1996), 370-377.

[5] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Trans. Information Theory*, 22 (1976), 472–492.

[6] A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature schemes, *Proc. of Crypto'86, Lecture Notes in Computer Science,* 263 (1987), 186–194.

[7] I. Ingemarsson, D.T. Tang, and C.K. Wong, A conference key distribution system, *IEEE Trans. Inform. Theory*, 28 (1982), 714–720.

[8] R. Julian and R. Abel, Difference Families, *The CRC Handbook of Combinatorial Designs*, eds. C. J. Colbourn and J. H. Dinitz, CRC Press, Boca Raton, Fla., (1996), 270–287.

[9] D. Jungnickel and A. Pott, Difference Sets: Abelian, *The CRC Handbook of Combinatorial Designs*, eds. C. J. Colbourn and J. H. Dinitz, CRC Press, Boca Raton, Fla., (1996), 297–307.

[10] K. Koyama and K. Ohta, Identity-based conference key distribution system, *Proc. of Crypto'87, Lecture Notes in Computer Science*, 293 (1987), 175–184.

[11] C.W.H. Lam, L. Thiel, and S. Swiercz, The non-existence of finite projective planes of order 10, *Canad. J. Math.*, 41 (1989), 1117–1123.

[12] T. Okamoto, Proposal for identity-based key distribution system, *Electron. Letters*, 22 (1986), 1283–1284.

[13] D.G. Sarvate and J. Seberry, Encryption methods based on combinatorial designs, *Ars Combinatoria*, 21A (1986), 235–246.

[14] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction of Computer Security*, Prentice-Hall, New York, 1988.

[15] J. Seberry and M. Yamada, Hadamard Matrices, Sequences and Block Designs, *Contemporary Design Theory: A Collection of Surveys*, eds. J. Dinitz and D. Stinson, J. Wiley, New York, (1992), 431- 560.

[16] A. Shamir, Intentity-based cryptosystems and signature schemes, *Proc. of Crypto'84, Lecture Notes in Computer Science*, 196 (1985), 47–53.

[17] A. Shimbo and S. Kawamura, Cryptoanalysis of several conference key distribution schemes, *Proc. of ASIACRYPT'91*, (1991), 155–160.

[18] R.M. Wilson, Cyclotomy and difference families in elementary abelian groups, *J. Number Theory*, 4 (1972), 17–47.