UMK_{Gm} TP: User Friendly Multi Group Key Transfer Protocol with Circulant Matrices

Shruti Nathani[1], B.P. Tripathi[2], S.K. Bhatt[3]

[1,2,3]Dept. of Mathematics, Govt. N.P.G. College of Science, Raipur (C.G.) India.

ABSTRACT

Most existing traditional group key distribution protocols are largely designed for a single group. They establish a single key for a single group. Many group oriented applications require multi-group key establishments at time. In which user may join multiple groups simultaneously. Recently, in 2018, C.F. Hsu et al. gave new type of user oriented multi-group key establishments using secret sharing (UMKESS). As many other group Key establishments schemes this protocol (UMKESS) is also polynomial based in which to distribute and recover the secret group key, the key generation centre(KGC) and each group member has to solve t-degree interpolating polynomial. Inspire from Hsu et al.'s UMKESS, in this paper, we present a new design of user friendly group key distribution protocol using secret sharing with circulant matrices. Because of using circulant matrices as a tool, our proposed protocol $UMK_{G_m}TP$ is become more efficient, secure and robust. Also, all the required security features of group communications are handle in $UMK_{G_m}TP$.

Key words: multi-group key establishment, secret sharing scheme, circulant matrices, key transfer protocol.

I INTRODUCTION

The traditional one to one communication has been expanded into one-to-many and many-to-many communication. This type of communications involving multiple users($n \geq 2$) are called group communication [11]. For a secure group communication a group key is needed to be shared among all the group members. That is, before exchanging communication messages a key establishment protocol must be used to construct the session keys for legitimate participants in the communication [19]. This session a key is then uses by the group users to communicate their secrets, to encrypt and decrypt sensitive information and to authenticate messages in the group.

The group key establishment protocols are often classified into two types:[2]

(a) Centralized, also called distributive group key establishment protocols, where a server is responsible for generate a group key and distribute the group key to all the group members. This type of protocols is also called GKT/GKD protocol.

(b) Distributed, also called, contributory group key establishment, in which there is no server, is required and group key is generated by the contribution of all the group members. This type is also known as group key agreement (GKA) protocol.

In the past few years a large amount of research work on group key transfer protocol has been published in the literatures. The most widely used group key transfer protocols are based on secret sharing scheme(SSS), which was first introduced by both Blakley[7] and Shamir[1], independently in 1979. Then the first group key transfer protocol using secret sharing scheme (SSS) is proposed in 1989 by Laih et al.[5]. Later, there are several other group key transfer protocols [8,9,10] following the same concept of using SSS was proposed.

In 2010, Harn et al.[10] proposed, a first authenticated GKT protocol based on SSS. The confidentiality and authentication of this novel GKT protocol is information theoretically secure. But, in this protocol, to distribute and recover the secret group key, KGC and each group member has to compute a t-degree interpolating polynomial. At the same time, many research articles [11,12,13,16,17] based on Harn et al.'s[10] authenticated protocol using SSS with the computation of a t-degree interpolating polynomial has been proposed.

To overcome, this drawback, in 2016, Hsu et al. [2] gave an efficient GKT protocol. In their scheme the information related to group keys was hidden by vandermonde matrix and to distribute the group key efficiently they employed linear secret sharing scheme on vandermonde matrix, which reduces the computation load of each group member.

Recently in 2018, S. Nathani et al.[14] also gave an authenticated and secure GKT protocol based on secret sharing scheme with circulant matrices. But all this above cited conventional GKT protocols can establish a single group key at a time, that is, establish a single group key for a single group.

With the rapid development of group oriented services such as business conferencing system, wireless body area network, programmable routey communications and file sharing tools etc, require more and more multi-group communications in which users may join multiple groups simultaneously.

Recently, a new type of user oriented multi-group key establishments using secret sharing (UMKESS) is proposed by C.F. Hsu et al.[3] in 2018. This multi-group key establishment scheme is also polynomial based. That means, again to distribute and recover the secret group key, KGC and each group member has to solve t degree interpolating polynomial.

Therefore, inspire from C.F. Hsu et al.'s [3], UMKESS protocol, we extend our conventional GKT protocol [14] into multi-group key transfer protocol on SSS with circulant matrices. In this paper, we propose a new design of user friendly multi-group key distribution protocol using SS with circulant matrices.

Some unique features of our protocol are summarized below:

- A circulant matrices based key distribution protocol for multi-group communications is proposed.
- We use circulant matrix as a tool and present an efficient computation of group keys. Since information related to group keys is a hidden using circulant matrix. Thus, each participating group member and KGC has to calculate only first row of the matrix. This gives us much less computational complexity.
- Each user keeps only one share with KGC at the time of registration and the share can be used to recover multiple group keys.
- In the whole proposed scheme, the group key is authenticated by each user of distinct groups and KGC. Also, authentication has been done by only one message in each group.
- The KGC can manage user joining or leaving dynamically. There has no rekeying overhead.
- All the required security features are handling in our proposed multi-group key transfer protocol.

II PRELIMINARIES

(a) Secret Sharing: In a secret sharing scheme, a secret S is divided into n shares and shared among a set of n shareholders by a mutually trusted dealer in such a way that authorized subset of shareholders can reconstruct the secret but unauthorized subset of share holders cannot determine the secret. If any unauthorized subset of shareholders cannot obtain any information about the secret, then the scheme is called perfect.[2]

(b) Circulant Matrix:[4] A Circulant matrix is a square matrix where, given the first row, the successive rows are obtained by cyclically right shifting the present row by one element. Thus the i^{th} row of a circulant matrix of size $(n \times n)$ is obtained by cyclically right shifting the $(i-1)^{th}$ row by one position, for $i = 2$ to n , given the first row. Let the first row be the row vector ,$[c(1), c(2), \ldots, c(n-1), c(n)]$. Then the circulant matrix C is obtained as

$$C = \begin{bmatrix} c(1) & c(2) & \cdots & c(n) \\ c(n) & c(1) & \cdots & c(n-1) \\ \vdots & \vdots & \cdots & \vdots \\ c(2) & c(3) & \cdots & c(1) \end{bmatrix}$$

The most important property of circulant matrices is they are multiplicatively commutative.

(c) SSS based on Circulant matrix for multi-group communications: Suppose a group of n participants $\{U_1, U_2, U_3, \cdots, U_n\}$ want to communicate in a secure multi-group communication with their long term secrets $\{x_1, x_2, \ldots x_n\}$ shared with only KGC. Also for multi-groups communication we have to take a batch of group $\{G_1, G_2, \ldots, G_m\}$ and a mutually trusted KGC. Actually this scheme consists of two algorithms [14].

(d) Secret generation algorithm: To form Circulant matrix for each user $U_j (1 \le i \le n)$ in each particular group $G_i (1 \le i \le m)$ KGC first picks the shared secret x_j of each user U_j and make circulant matrix $[C_{ij}]$ as below :

$$[C_{ji}] = \begin{bmatrix} c(1) & c(2) & \cdots & c(n) \\ c(n) & c(1) & \cdots & c(n-1) \\ \vdots & \vdots & \cdots & \vdots \\ c(2) & c(3) & \cdots & c(1) \end{bmatrix}$$
$$= Circ(x_j^1, x_j^2, \ldots \ldots \ldots, x_j^m)$$

where $1 \le j \le n$

and m denotes the number of group users in each particular group G_i and then calculate the secrets of S_{ji} of each user $U_j (1 \le j \le n)$ by computing

$$S_{ji} = [C_{ji}] * Circ(r_{1i}, r_{2i}, \ldots \ldots, r_{ji})$$
$$\text{for } 1 \le j \le n, 1 \le i \le m$$

Thus, this algorithm outputs with a list of secret shares $S_{ji} (1 \le j \le n, 1 \le i \le m\ )$.

(e) Secret Reconstruction Algorithm: This algorithm takes all the shares $S_{ji}(1 \le j \le n, 1 \le i \le m)$ each participating member U_j has long term private key x_j

and public vector $\underset{r\ ji}{\rightarrow} = (r_{1i}, r_{2i}, \dots \dots, r_{ni})$ as inputs and outputs the secret

$$S = s_1 + s_2 + \cdots + s_n$$

by computing each product

$$\boldsymbol{S_{ji} = Circ(x_j^1, x_j^2, \dots \dots, x_j^m). Circ(r_{1i}, r_{2i}, \dots, r_{ji})}$$

$$\boldsymbol{(for\ 1 \le j \le n, 1 \le i \le m).}$$

III PROPOSED PROTOCOL

We suppose that there are n users $\{U_1, U_2, \dots \dots, U_n\}$ participated in multi-group communications. Each user is required to register itself at KGC and KGC keeps tracking all the registered group member which includes removing any unsubscribed group participants or adding new member. To achieve secure multi-group communications, KGC has to selects multi-group session keys for all the running groups simultaneously and securely distributes these keys to all the valid registered members of particular groups. Therefore, the only valid members who belong to that particular group can easily derive this group's session key.

The proposed group key transfer protocol for multi-group communications consist of three phases: Initialization, user registration, multi group key distribution and establishment. Here we assume that there are n users $\{U_1, U_2, \dots \dots, U_n\}$ participated in multi-group communications denoted by $\{G_1, G_2, \dots \dots, G_m\}$.

(a) Initialization: The KGC selects a safe large prime p, and a secure one way hash function $h(.)$ whose domain is GF(p). The KGC publishes p and $h(.)$.

(b) User Registration: Each user is required to register at the KGC for subscribing the key distribution service. The KGC keeps tracking all the registered users or adding new users. During the registration each user $U_j(1 \le j \le n)$ shares his/her long term secret $x_j \in K$, $(1 \le j \le n)$ with KGC in a secure manner.

(c) Multi-group key generation, distribution and establishment: Suppose a group of n members $\{U_1, U_2, \dots \dots, U_n\}$ want to communicate in a secure multi-group communication with their long term secrets $\{x_1, x_2, \dots \dots, x_n\}$ shared with only trusted party KGC secretly. Here we also assume a batch of groups $\{G_1, G_2, \dots \dots, G_m\}$ which are handle by KGC simultaneously. The process of multigroup key generation, distribution and establishment contain five steps:

(i) Step 1: The initiator sends a key generation request to KGC for multiple groups with a list of groups $\{G_1, G_2, \dots \dots, G_m\}$ and each group is represented as $G_i = \{U_1, U_2, \dots \dots, U_j\}$, $1 \le i \le m\ where\ j \in \{1,2, \cdots, n\}$.

(ii) Step 2: KGC finally broadcast the list of all groups $\{G_1, G_2, \dots \dots, G_m\}$ to all members as a response.

(iii) Step 3: For each group member $U_j, 1 \le j \le n$, he/she decides to join more than one groups $G_i(1 \le i \le m)$ simultaneously. Then each group user sends their random value r_{ji}, (for $1 \le j \le n$, $1 \le i \le m$) for each group G_i in which they want to join.

(iv) Step 4: Now KGC received all the random values send by all the group participants $U_j, (1 \le j \le n)$. Then KGC broadcast the actual list of participants of each particular group according their random values sent by each group user. This list of number of participants in each particular group helps the group participants to make circulant matrices.

(v) Step 5: Now KGC randomly selects the group keys $K_{G_i}(1 \le i \le m)$ for all the groups $G_i(1 \le i \le m)$. Then KGC compute the secrets $S_j(1 \le j \le m)$ of each user U_j in each particular group $G_i(1 \le i \le m)$ by computing the product

[Circulant matrices of shared secrets of each user $\boldsymbol{U_j}$ in the group $\boldsymbol{G_i}$]* [Circulant matrix of random values $\boldsymbol{r_{ji}}$ of each user $\boldsymbol{U_j}$ in the group $\boldsymbol{G_i}$] =$\boldsymbol{s_{ji}}$. $\boldsymbol{(1 \le i \le m,\ 1 \le j \le n)}$

$[C_{ji}] * Circ(r_{1i}, r_{2i}, \dots \dots, r_{ji}) = s_{ji}$

Here, m denotes the number of members in the group G_i. After this computation of secret of each user U_j in particular groups, KGC also computes some additional values $u_{ji} = S_i - s_{ji}$, where

$S_i = Circ(K^1{}_{G_i}, K^2{}_{G_i}, \dots \dots, K^j{}_{G_i})$,

for $1 \le j \le n, 1 \le i \le m$ and

$$Auth_i \quad = h(K_{G_i}, U_1, U_2, \dots, U_j, r_{1i}, r_{2i}, \dots, r_{ji}, u_{1i}, u_{2i}, \dots, u_{ji})$$

for , $1 \le j \le n, 1 \le i \le m$.

At last, finally KGC broadcast $(Auth_i, (u_{ji})_{G_i}$ for $1 \le i \le m, 1 \le j \le n$.

Here, i represents number of groups and j represents number of participants in each group G_i.

(vi) Step: 6 Now each participating group member $U_j, 1 \le j \le n$, knowing their corresponding public value u_{ji}, in each particular group $G_i, (1 \le i \le m)$, is able to compute the product

$$[C_{ij}] * Circ(r_{1i}, r_{2i}, \dots \dots, r_{ji}) = s_{ji}$$

and recover the group key K_{G_i} by computing,

$$S_i = (u_{ji} + s_{ji})$$

Which is of the form

$$S_i = Circ(K_{G_i}^1, K_{G_i}^2, \ldots\ldots K_{G_i}^j)$$

(for , $1 \le j \le n$, $1 \le i \le m$)

Afterwards, each u_{ji}, ($for\ 1 \le j \le n,\ 1 \le i \le m$) authenticates their corresponding groups $G_i$by computing

$Auth_i^* = h(K_{G_i,} U_1, U_2, \ldots, U_j, r_{1i}, r_{2i}, \ldots, r_{ji}, u_{1i}, u_{2i}, \ldots u_{ji})$

for $1 \le j \le n,\ 1 \le i \le m$

and then checks this value by

$$Auth_i = Auth_i^*.$$

If this result is correct then each participant $U_j(1 \le j \le n)$, in the group $G_i(1 \le i \le m)$ authenticates the group key K_{G_i} is sent from KGC.

IV AN EXAMPLE

In our example we assume a group of 7 members $\{U_1, U_2, U_3, U_4, U_5, U_6, U_7\}$ want to generate a secure group communications in multiple groups simultaneously.

(a) User Registration: During registration each user U_j, $1 \le j \le 7$, shares his/her long term secrets $x_i \in K$ with KGC. Suppose U_1 Shares $x_1 = 2$, U_2 Shares $x_2 = 1$, U_3 Shares $x_3 = 4$, U_4 Shares $x_4 = 3$, U_5 Shares $x_5 = 10$, U_6 Shares $x_6 = 5$, U_7 Shares $x_7 = 7$ in a secure manner. KGC publishes $h(\cdot)$.

(b) Group Key Generation and Distribution:

In our example we assume a batch of groups$\{G_1, G_2, G_3\}$, in which there 7 group members want to join simultaneously.

Step 1: Suppose $U_2(initiator)$ sends a key generation request to KGC with a list of groups $\{G_1, G_2, G_3\}$.

Step 2: KGC broadcast the list of groups $\{G_1, G_2, G_3\}$ to all members as a response.

Step 3: Here each group member U_j, $(1 \le j \le 7)$, he/she decides to join more than one groups G_i, ($1 \le i \le 3$). Then each group participants sends their radom values r_i, for each group G_i in which they want to join.

Suppose ,U_1 sends $r_{11} = 2$, $r_{13} = 1$, U_2 sends $r_{21} = 1$, $r_{22} = 8$, U_3 sends $r_{32} = 2$, U_4 sends $r_{41} = 10$, $r_{43} = 3$, U_5 sends $r_{51} = 11$, $r_{52} = 6$, U_6 sends $r_{61} = 4$, $r_{62} = 2$, U_7 sends $r_{73} = 9$ to KGC.

Step 4: Now KGC received all the random keys send by the 7 users $\{U_1, U_2, U_3, U_4, U_5, U_6, U_7\}$.

Then, KGC broadcast the actual list of participants $U_j(1 \le j \le 7)$ of each particular group $G_i(1 \le i \le 5)$. That means KGC broadcast

$$(\{U_1, U_2, U_4, U_5, U_6,\} \in G_1,$$

$\{U_2, U_3, U_5\} \in G_{2,}$ $\{U_1, U_4, U_7\} \in G_3$) list of all group members publicly.

Step 5: Now KGC randomly selects the 3 group keys $K_1 = 100$, $K_2 = 200$, $K_3 = 50$, to all the 3 groups $\{G_1, G_2, G_3\}$.

Now KGC compute the secrets s_j of each user U_j of each particular groups $G_i(1 \le j \le 7,\ 1 \le i \le 3)$.

For this KGC, first has to make the circulant matrices of each participating group user $U_j(1 \le j \le 7)$ in each particular group $G_i(1 \le i \le 3)$, with the help of their corresponding shared secret values.

$$x_1 = 2, x_2 = 1, x_3 = 4, x_4 = 3, x_5 = 10, x_6 = 5, x_7 = 7$$

That means, for G_1, $\{U_1, U_2, U_4, U_5, U_6,\}$,

$$C_{11} = Circ(2^1, 2^2, 2^3, 2^4, 2^5) = Circ(2,4,8,16,32)$$

$$C_{21} = Circ(1^1, 1^2, 1^3, 1^4, 1^5) = Circ(1,1,1,1,1)$$

$$C_{41} = Circ(3^1, 3^2, 3^3, 3^4, 3^5) = Circ(3,9,27,81,243)$$

$$C_{51} = Circ(10^1, 10^2, 10^3, 10^4, 10^5) = Circ(10,100,1000,10000,100000)$$

$$C_{61} = Circ(5^1, 5^2, 5^3, 5^4, 5^5) = Circ(5,25,125,625,3125)$$

Then, $s_{11} = [C_{11}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$

$$= Circ(2,4,8,16,32) * Circ(2,1,10,11,4)$$

$$= Circ(300,538,446,230,212).$$

$$s_{21} = [C_{21}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$

$$= Circ(1,1,1,1,1) * Circ(2,1,10,11,4)$$

$$= Circ(28,28,28,28,28).$$

$$s_{41} = [C_{41}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$

$$= Circ(3,9,27,81,243) * Circ(2,1,10,11,4)$$

$= Circ(1392,3450,3090,1284,948)$.

$$s_{51} = [C_{51}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$

$$= Circ(10,100,1000,10000,100000) * Circ(2,1,10,11,4)$$

$$= Circ(211420,1114210,1142200,$$

$$422110,221140)$$

$$s_{61} = [C_{61}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(5,25,125,625,3725) \quad * Circ(2,1,10,11,4)$$
$$= Circ(11460,44680,43800,$$
$$16580,9620)$$

For group G_2, $\{U_2, U_3, U_5\}$,

$$C_{22} = Circ(1^1, 1^2, 1^3) = Circ(1,1,1).$$

$$C_{32} = Circ(4^1, 4^2, 4^3) = \quad Circ(4,16,64).$$
$$C_{52} = Circ(10^1, 10^2, 10^3) = \quad Circ(10,100,1000).$$

Then,

$$s_{22} = [C_{22}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(1,1,1) * Circ(8,7,6)$$
$$= Circ(21,21,21).$$
$$s_{32} = [C_{32}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(4,16,64) * Circ(8,7,6)$$
$$= Circ(576,540,648).$$
$$s_{52} = [C_{52}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(10,100,1000) * Circ(8,7,6)$$
$$= Circ(7680,6870,8760).$$

For group G_3, $\{U_1, U_4, U_7\}$,

$$C_{13} = Circ(2^1, 2^2, 2^3) = Circ(2,4,8).$$

$$C_{43} = Circ(3^1, 3^2, 3^3) = \quad Circ(3,9,27).$$
$$C_{52} = Circ(7^1, 7^2, 7^3) = \quad Circ(7,49,343).$$

Then,

$$s_{13} = [C_{13}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(2,4,8) * Circ(1,3,9)$$

$$= Circ(62,82,38).$$
$$s_{43} = [C_{43}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(3,9,27) * Circ(1,3,9)$$
$$= Circ(165,261,81).$$
$$s_{73} = [C_{73}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(7,49,343) * Circ(1,3,9)$$
$$s_{73} = Circ(1477,3157,553).$$

Now, KGC computes the five additional values for group G_1,

$$u_{11} = S - s_{11}$$
$$u_{11} = Circ(100^1, 100^2, 100^3, 100^4, 100^5) \; - Circ(300,538,446,230,212).$$
$$= Circ(-200,9462,999554,99999770,$$
$$9999999788).$$

$$u_{21} = S - s_{21}$$
$$u_{21} = Circ(100^1, 100^2, 100^3, 100^4, 100^5) \; - Circ(28,28,28,28,28).$$
$$= Circ(72,9972,999972,99999972,$$
$$9999999972).$$

$$u_{41} = S - s_{41}$$
$$u_{41} = Circ(100^1, 100^2, 100^3, 100^4, 100^5) \; - Circ(1392,3450,3090,1284,948).$$
$$= Circ(-1292,6550,996910,99998716$$
$$,9999999052).$$

$$u_{51} = S - s_{51}$$
$$u_{51} = Circ(100^1, 100^2, 100^3, 100^4, 100^5) \; - Circ\begin{pmatrix}211420, 1114210, 1142200, \\ 422110, 221140\end{pmatrix}.$$

$= Circ(-211320, -1104210, -142200,$
$99577890, 9999778890).$

$$u_{61} = S - s_{61}$$
$$u_{61} = Circ(100^1, 100^2, 100^3, 100^4, 100^5) - Circ(11460, 44680, 43800, 16580, 9620).$$
$$= Circ(-11360, -34680, 956200, 99983420,$$
$9999990380).$
and the value of
$Auth_1 = h(K_{G_1} = 100, \{U_1, U_2, U_4, U_5, U_6\}, r_{11}, r_{21}, r_{41}, r_{51}, r_{61}, u_{11}, u_{21}, u_{41}, u_{51}, u_{61})$.

KGC computes three additional values for group G_2,

$$u_{22} = S - s_{22}$$
$$u_{22} = Circ(200^1, 200^2, 200^3) - Circ(21,21,21)$$
$$= Circ(179, 39979, 7999979).$$

$$u_{32} = S - s_{32}$$
$$u_{32} = Circ(200^1, 200^2, 200^3) - Circ(576, 540, 648)$$
$$= Circ(-376, 39460, 7999352).$$

$$u_{52} = S - s_{52}$$
$$u_{52} = Circ(200^1, 200^2, 200^3) - Circ(7680, 6870, 8760)$$
$$= Circ(-7480, 33130, 7991240).$$
and the value of
$Auth_2 = h\left(K_{G_2} = 200, \{U_2, U_3, U_5\}, r_{22}, r_{32}, r_{52}, u_{22}, u_{32}, u_{52}\right).$

Also, KGC has to compute 3 additional values for group $G_3 \in \{U_1, U_4, U_7\}$.

$$u_{13} = S - s_{13}$$
$$u_{13} = Circ(50^1, 50^2, 50^3) - Circ(62, 82, 38)$$
$$= Circ(-12, 2418, 124962).$$
$$u_{43} = S - s_{43}$$
$$u_{43} = Circ(50^1, 50^2, 50^3) - Circ(165, 261, 81)$$
$= Circ(-115, 2239, 124919).$
$$u_{73} = S - s_{73}$$
$$u_{73} = Circ(50^1, 50^2, 50^3) - Circ(1477, 3157, 553)$$
$$= Circ(-1427, -657, 124447).$$
and the value of
$$Auth_3 = h\left(K_{G_3} = 50, \quad \{U_1, U_4, U_7\}, \quad r_{13}, r_{43}, r_{73}, u_{13}, u_{43}, u_{73}\right).$$
Thus, KGC finally broadcast,
$$\{Auth_1, Auth_2, Auth_3, \{u_{11}, u_{21}, u_{41}, u_{51}, u_{61}\}_{G_1},$$
$\{u_{22}, u_{32}, u_{52}\}_{G_2}, \{u_{13}, u_{43}, u_{73}\}_{G_3}\}.$

Step 6: At last to compute the common group key, each participating group members of group,
$$G_1 \in \{U_1, U_2, U_4, U_5, U_6\}, \quad G_2 \in \{U_2, U_3, U_5\},$$
$$, \; G_3 \in \{U_1, U_4, U_7\},$$
has to solve the equation
$$S = (u_{ji} + s_{ji})$$
where, $S = Circ(K_i^1, K_i^2, \dots, K_i^j)$
here, j denotes the number of participants in the group i .
Therefore, for group G_1,
User U_1 , computes
$$s_{11} = [C_{11}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(2,4,8,16,32) * Circ(2,1,10,11,4)$$
$$= Circ(300, 538, 446, 230, 212).$$
So, $S = u_{11} + s_{11}$
S = $Circ(-200, 9462, 999554,$
$99999770, 9999999788) + \quad Circ(300, 538, 446, 230, 212)$
S=Circ(100,10000,1000000,100000000,10000000000)
S=Circ(100,$100^2, 100^3, 100^4, 100^5$)

Thus, G_{K_1} = 100.

$$s_{21} = [C_{21}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(1,1,1,1,1) * \ Circ(2,1,10,11,4)$$
$$= Circ(28,28,28,28,28).$$

So, $S = u_{21} + s_{21}$

S= $Circ(72,9972,999972,$
$99999972, 9999999972) + \quad Circ(28,28,28,28,28)$
=Circ(100,10000,1000000, 100000000,10000000000)
S = Circ(100,$100^2, 100^3, 100^4, 100^5$)
Thus, G_{K_1} = 100.

$$s_{41} = [C_{41}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(3,9,27,81,243) * Circ(2,1,10,11,4)$$
$$= Circ(1392,3450,3090,1284,948).$$

So, $S = u_{41} + s_{41}$
S = $Circ(-1292,6550,996910,$
$99998716,9999999052) + \quad Circ(1392,3450,3090,1284,948)$
S=Circ(100,10000,1000000, 100000000,10000000000)
S =Circ(100,$100^2, 100^3, 100^4, 100^5$)
Thus, G_{K_1} = 100.

$$s_{51} = [C_{51}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(10,100,1000,10000,100000) \quad * Circ(2,1,10,11,4)$$
$$= Circ(211420,1114210,1142200422110,221140).$$

So, $S = u_{51} + s_{51}$
S = $Circ(-211320, -1104210,$
$-142200,99577890,9999778860) + \ Circ(211420,1114210,1142200,422110,221140).$
S=Circ(100,10000,1000000, 100000000,10000000000)
S=Circ(100,$100^2, 100^3, 100^4, 100^5$).
Thus, G_{K_1} = 100.

$$s_{61} = [C_{61}] * Circ(r_{11}, r_{21}, r_{41}, r_{51}, r_{61})$$
$$= Circ(5,25,125,625,3725) \quad * Circ(2,1,10,11,4)$$
$$= Circ(11460,44680,43800, 16580,9620).$$

So, $S = u_{61} + s_{61}$
S = $Circ(-11360, -34680,956200,$
$99983420,9999990380) +$
$Circ(11460,44680, 43800,$
$16580,9620)$
S=Circ(100,10000,1000000,100000000,10000000000)
S=Circ(100,$100^2, 100^3, 100^4, 100^5$)
Thus, G_{K_1} = 100.

Hence, all the group users of group G_1 gets the group key K_{G_1} = 100.

For, group $G_2 \in \{U_2, U_3, U_5\}$,
User U_2 computes,

$$s_{22} = [C_{22}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(1,1,1) * Circ(8,7,6)$$
$$= Circ(21,21,21).$$

So, $S = u_{22} + s_{22}$
S = $Circ(179,39979,7999979) + \ Circ(21,21,21)$
S=Circ(200,40000,8000000)
S=Circ(100,$200^2, 200^3$)
Thus, G_{K_2} = 200.
User U_3 computes,

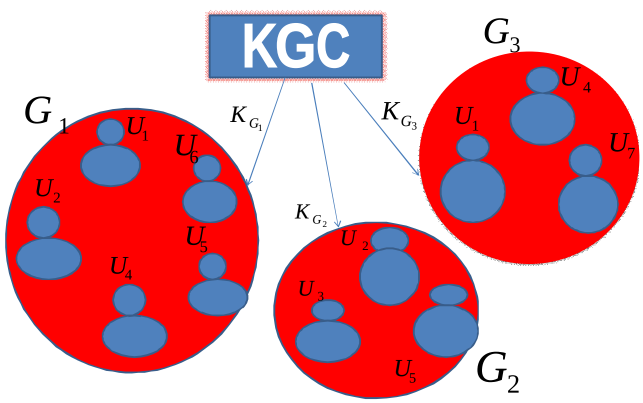
$$s_{32} = [C_{32}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(4{,}16{,}64) * Circ(8{,}7{,}6)$$
$$= Circ(576{,}540{,}648).$$

So, $S = u_{32} + s_{32}$
S= $Circ(-376{,}39460{,}7999352)$+$Circ(576{,}540{,}648)$
S=Circ(200,40000,8000000)
S=Circ(200,200^2, 200^3).
Thus, $G_{K_2} = 200$.
User U_5 computes,

$$s_{52} = [C_{52}] * Circ(r_{22}, r_{32}, r_{52})$$
$$= Circ(10{,}100{,}1000) * Circ(8{,}7{,}6)$$
$$= Circ(7680{,}6870{,}8760).$$

So, $S = u_{52} + s_{52}$
S= $Circ(-7480{,}33130{,}7991240)$ + $Circ(7680{,}6870{,}8760)$.

S = Circ(200,40000,8000000)
S = Circ(200,200^2, 200^3).
Thus, $G_{K_2} = 200$.
Hence, all the group users of group G_2 gets the group key $K_{G_2} = 200$.



For, group $G_3 \in \{U_1, U_4, U_7\}$,
User U_1 computes,

$$s_{13} = [C_{13}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(2{,}4{,}8) * Circ(1{,}3{,}9)$$

$= Circ(62{,}82{,}38)$.
So, $S = u_{13} + s_{13}$
S = $Circ(-12{,}2418{,}124962)$+ $Circ(62{,}82{,}38)$.
S=Circ(50,2500,125000)
S=Circ(50,50^2, 50^3)
Thus, $G_{K_3} = 50$.
User U_4 computes,

$$s_{43} = [C_{43}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(3{,}9{,}27) * Circ(1{,}3{,}9)$$

$= Circ(165{,}261{,}81)$.
So, $S = u_{43} + s_{43}$
S = $Circ(-115{,}2239{,}124919) + Circ(165{,}261{,}81)$.
S=Circ(50,2500,125000)
S=Circ(50,50^2, 50^3)
Thus, $G_{K_3} = 50$.

$$s_{73} = [C_{73}] * Circ(r_{13}, r_{43}, r_{73})$$
$$= Circ(7{,}49{,}343) * Circ(1{,}3{,}9)$$
$$s_{73} = Circ(1477{,}3157{,}553).$$

User U_7 computes,
So, $S = u_{73} + s_{73}$

S= $Circ(-1427, -657, 124447)$ + $Circ(1477, 3157, 553)$.
S=Circ(50,2500,125000)
S=Circ(50,$50^2, 50^3$)
Thus, $G_{K_3} = 50$.
Hence, all the group users of group G_3 gets the group key $K_{G_3} = 50$.

V SECURITY ANALYSIS

Theorem: The proposed protocol possesses key freshness, key confidentiality and key authentication.

Proof: Key Freshness: In our proposed protocol for each new communication session m new group keys $\{G_{K_1}, G_{K_2}, \dots\dots, G_{K_m}\}$ associated with $\{G_1, G_2, \dots\dots, G_m\}$ are randomly selected by KGC for each multi-group key service request. Also, to compute the group key $K_{G_i}(1 \le i \le m)$ each group user $U_j(1 \le j \le n)$ has to calculate

$S = (u_{ji} + s_{ji})$, where

$$s_{ji} = [C_{ij}] * Circ(r_{1i}, r_{2i}, \dots\dots, r_{ji})$$

$$s_{ji} = (x_j^1, x_j^2, \dots\dots, x_j^m) * Circ(r_{1i}, r_{2i}, \dots\dots, r_{ji})$$

Which is a function of shared secrets of each user U_j and random challenges(public values) $r_{ji}(1 \le j \le n, 1 \le i \le m)$ selected by each group member $U_j(1 \le j \le n)$ for each new communication service request. Thus, it is obvious that the group key K_{G_i} will be fresh that is new and different for each new communication session.

Key Confidentiality: Key secrecy is provided due to the security feature of SSS based on circulant matrices for multiple groups. To handle multiple groups at a time KGC has to select multiple group keys $\{K_{G_1}, K_{G_2}, \dots\dots, K_{G_m}\}$, the respective group members have calculate

$$\boldsymbol{S_i = (u_{ji} + s_{ji})(= Circ[K_{G_i}^1, K_{G_i}^2, \dots\dots, K_{G_i}^t])}$$

Where, u_{ji} are the public values sent by KGC and

$$\boldsymbol{s_{ji} = [C_{ji}] * Circ(r_{1i}, r_{2i}, \dots\dots, r_{ji})}$$
$$\boldsymbol{s_{ji} = (x_j^1, x_j^2, \dots\dots, x_j^t) * Circ(r_{1i}, r_{2i}, \dots\dots, r_{ji})}$$

Where t denotes the number of members in the group G_i. This shared secret value s_{ji} assured that only authorized group member is able to recover the group key K_{G_i} which is of the form

$$\boldsymbol{S_i = Circ(K_{G_i}^1, K_{G_i}^2, \dots\dots, K_{G_i}^t)}$$

where t represent the number of members in the group G_i.
Hence, key confidentiality is surely achieved in our proposed scheme.

Key Authentication: In key distributing phase, the KGC also compute $Auth_i$ for all the multiple groups G_i simultanously. Also, each user U_j authenticates their corresponding groups G_i by computing

$$Auth^*i = h(K_{G_i}, U_1, U_2, \dots, U_j, r_{1i}, r_{2i}, \dots, r_{ji}, u_{1i}, u_{2i}, \dots, u_{ji})$$

for , $1 \le j \le n, 1 \le i \le m$.
and then check this hash value by $Auth_i = Auth_i^*$.
Also this key authentication is done only by one message for each group G_i.

Theorem(Insider attack): The proposed protocol **UMK$_{\mathbf{G_m}}$TP** is secure against insider attack.
Proof: At the time of registration, each participating group member U_j shared his/her long term secret key x_j only with KGC (a trusted authority). For each new communication session a new group key K_{G_i} is selected by KGC and makes some values $u_{ji} = (S - s_{ji})(1 \le i \le m, 1 \le j \le n)$ publicly known. Then each authorized group member knows their shared secret x_j with KGC and public values u_{ji} is able to compute the group key K_{G_i} which is of the form

$$S = Circ(K_G^1, K_G^2, \cdots, K_G^t).$$

Since , $S = u_i + s_i$,
where ,

$$\mathbf{s_{ji} = (x_j^1, x_j^2, \dots\dots, x_j^t) * Circ(r_{1i}, r_{2i}, \dots\dots, r_{ji})}$$

Therefore, the secret $x_j \in K$ of each group member shared with KGC remains unknown to outsiders and also each authorized group member is able to recover the group key but not able to obtain other member's long term secret x_j. Thus, our proposed protocol resist against insider attack.

Theorem (Forward and Backward Secrecy): The proposed protocol $\mathbf{UMK_{G_m}TP}$ provide backward and forward secrecy, that is newly joined members cannot recover the old group keys and those old members who left the group cannot access the current group key.

Proof: In our proposed $\mathbf{UMK_{G_m}TP}$ protocol, for every multi-group session, if new members join in or old members left from groups, the KGC needs to distribute new group keys to all existing group members. In each group the group key K_{G_i} is derived from the current group members long term secrets x_j's and fresh random challengesr_{ji}. Also, our whole computation is totally depends on the number of members in the current group. Thus, the newly joined members can recover the current group key but cannot recover the previous group keys and those old members who left the group cannot recover the current group key. Thus, our protocol achieves both forward and backward secrecy of group communication.

VI CONCLUSION

We defined a new type of, circulant matrices based key transfer protocol for multi-group communications. Because of using circulant matrices as a tool, our proposed multi-group key transfer protocol takes much less time than other existing multi-group key transfer protocols. Also all the required security attributes are addressed in detail and the confidentiality of our proposed protocol is unconditionally secure.

REFERENCES

[1] A. Shamir, "How to share a secret ", Commun. ACM vol. 22, no. 11, pp. 612-613, Nov. (1979).

[2] C.F. Hsu, L. Harn, Y. Mu, M. Zhang, X. Zhu, "Computation efficient key establishment in wireless group communications ", wireless network , vol. 23, PP. 289-297, (2016).

[3] C.F. Hsu, L. Harn, B. Zeng, "UMKESS: user oriented multigroup key establishments using secret sharing", wireless networks ,(2018).

[4] C. Rajarama, J. N. Sugatoor, T. Y. Swamy, " Diffie-Hellman type key exchange, ElGamal like ecryption/decryption and proxy re-encryption using circulant matrices ", International Journal of Network Security, vol. 20, no. 4, PP. 617-624, July (2018).

[5] C.S. Laih and J. Y. Lee, "A new threshold scheme and its applications in designing the conference key distribution cryptosystem", Inf. Process. Lett., vol. 32, no. 3, PP. 95-99, (1989).

[6] C.Y. Lee, Z.H. Wang, L.Harn , C.C. Chang, "Secure key transfer protocol based on secret sharing for group communications", IEICE Trans. Inf. & Syst. , vol. E94-D, no. 11, (2011).

[7] G. R. Blakely, "Safegaurding cryptographic keys" , in proc. AFIPS 1979, National Computer Conference, PP. 313-317. AFIPS, (1979).

[8] G. Saze, "Generation of key predistribution schemes using secret sharing schemes ", Discrete Applied Mathematics , vol. 128, PP. 239-249, (2003).

[9] L. Ch, J. Pieprzyk, "Conference key agreement from secret sharing ", Proc. Fourth Australasian Conf. Information Security and Privacy(ACISP'99), PP. 64-76, (1999).

[10] L. Harn, C. Lin, "Authenticated group key transfer protocol based on secret sharing", IEEE Trans. Comuter , vol. 59, no. 6, PP. 842-846, (2010).

[11] L. Harn, G. Gong, "Conference key establishment protocol using a multivariate polynomial and its applications", Security and Communication Networks, vol. 8, no. 9, PP. 1794-1800,(2014).

[12] L. Harn, C. Lin, "Efficient group Diffie-Hellman key agreement protocols", Comput. Elect. Eng., (2014).

[13] R. F. Olimid, "Cryptanalysis of a password based group key exchange protocol using secret sharing", Appl. Math. Inf. Sci., vol. 7, no. 4, PP. 1585-1590,(2013).

[14] S. Nathani, B.P. Tripathi, " An authenticated and secure group key transfer protocol with circulant matrices", Journal of Computer and Mathematical Sciences , vol. 9, no.12, PP. 2086-2095, (2018).