Multiple Operations for Secure Multicast Communication

Manisha Manjul
School of ICT
Gautam Buddha University
Greater Noida, U.P., India

Rajesh Mishra
School of ICT
Gautam Buddha University
Greater Noida, U.P., India

ABSTRACT

Group key management is a critical task to make the secure multicast application. Group manager is responsible for changing and re-distributing (rekeying) the group key whenever it deems necessary. Many applications will require a security infrastructure that ensures multiple levels of access control for group members and also require very fast rekeying so that it is not disruptive to their performance. We use a multi-operation key management scheme that achieves hierarchical group access control. Particularly, we use an integrated key graph that maintains keying material for all members with different access privileges. It also incorporates new functionalities that are not present in conventional multicast key management, such as user relocation on the key graph. We use the ECC-GKM elliptic curve cryptosystem for the key exchange and we show the multiple operations such as joining, leaving and switching for multiple levels of access control.

General Terms

Computer Network, Multicast Communication, Multicast Security

Keywords

Multicast, Group Key Management, Access Control, Elliptic Curve Cryptography

1. INTRODUCTION

Multicast communication is a scheme of wireless communication [11] for transmitting data from a single sender to multiple receivers or by a multiple sender to multiple receivers. Presently multicast is widely used in various applications [12] like video conference, IP communication, real time applications, distance learning etc. In multicast operation at a time so many members of same group or from a different group communicate with each other to convey the information. So to secure the communication [9] security is required in group communication at the time of every stage like group creation, member join, member leave etc. The security issues in multicast systems are also urgent to be addressed and the group key management is the foundation of multicast security. Although Unicast communication is predominant so far, the demand for group oriented communication has increased rapidly .So the demand for security in group communication has become a focus of attention and research [8] [10].

Access control is one of the most important issues among all other. Access control is achieved by encryption of data by using a key called as session key which is allotted to all the users. Groups can be dynamic and as well as static but widely used is dynamic so that members can join or leave the group. At each join and leave operation the key must be updated to provide forward as well as backward secrecy and managed by group key management schemes [14][15]. So this provides basic security requirements of any system i.e confidentiality, integrity and authentication to the group communication.

Traditional key management schemes are unable to solve issues for multiple services. For multiple services traditional schemes becomes inefficient. In these schemes are centralized, decentralized and distributed. This paper proposes multiple group key management that resolve the general problem of hierarchical access control. It also provides new features that are not in traditional schemes.

2. RELATED WORK

The Key management can be based on three approaches and these are centralized, distributed, hybrid schemes.

2.1 Centralized Scheme: It [2][5] has a central entity to govern all the keying process called “key distribution center” which generates and provide encryption keys to all the members. For the security requirement all other member has to trust on this entity. The main advantages of these schemes are as follows:

1. This scheme is easy to manage, since trust is provided by single entity.
2. It reduces some communication overheads.
3. A single entity is sufficient to deploy trust in whole group.

2.2 Decentralized Scheme: In this scheme [3][6] a group is divided in the sub group and task of key management is divided among these sub groups. There is no central entity to govern all these key management schemes. In this each member of group are equaled responsible for trust and perform the function of rekeying. Advantage of this scheme is that this is more flexible. Disadvantage of this schemes are (a) it does not always scale well, since distribution of management tasks across larger multicast groups can be complex. (b) The messages exchanged between group members can be prohibitively large in large networks.

2.3 Hybrid Scheme: It is combination of both the schemes [4] It implies distributed hierarchy of trust for providing rekeying and revoking key management [13]. For example if there is multilevel multicast group communication so at each level few of the members are selected to manage the sub entities at the second level members. Now the sublevel can govern key management of another level and this process goes on up to the bottom level. In this the properties of both the previous schemes can be fine tune by this as per our requirements.

A centralized key management uses logical tree structure to coordinate. In this key distributed center (KDC) [7] maintains the key tree .in this each node is a user private key, group key, key encrypted key. This faces some problem of overheads so later on a new scheme is proposed to reduce the overheads and calculate the rekey efficiently by using “one way function tree”. In this there is no KDC to generate a group key and key generation is based on the one way function. This reduces overhead from (2logn) to (log n).

Now in the place of one way function tree we can use pseudo random function to build and maintain rekeying process and provide key management. Few other schemes are based on Diffe-Hellman algorithm to provide key management, but this faces the problem of algorithm size and weight which use the tree based approach as our basic blocks to provide access control and to provide key management in multiple multicast group communications.

3. PROPOSED SOLUTION

The proposed group key management approach is based on Elliptic Curve Cryptography (ECC) [1]. This approach uses the ECC for the secure group key management in multi-level access control. One main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA. The implication of using a singular elliptic curve in cryptography is that the elliptic curve discrete logarithm problem on such curve is computed in polynomial time. A cryptographic algorithm with polynomial time complexity is said to be insecure because the algorithm can be easily broken without much computational difficulty. Nevertheless, other crucial decisions when implementing an efficient elliptic curve over GF (p) are deciding which coordinate system to use and which point multiplication method yields better performance. So, the system designer has the responsibility of making a wise choice for optimum system performance.

3.1 System Setup

Prior to commencement of communication, a trusted member is pre-assigned as the group manager GM. The GM generates the elliptic curve Ep(a, b), base point G=($x_{G,}$ y_G) and field size q for the group through the key server. The GM also fixes n, the maximum number of members the group can accommodate. The key server serves as the key distribution Centre (KDC) as well as providing reliable ordered message delivery within the group. The KDC shares the generated E_p(a, b) as well domain parameters G and q with every member of the group. Thus, these are announced publicly.

To formulate the hierarchical access problem, the ways of encrypting multiple data streams need to be clarified first. In the hierarchical access control scenario, there are two ways to encrypt and distribute multicast data. In the first method, resources are encrypted using separate keys, which are called Data Group Keys. The data group key used to encrypt resource r_i , denoted by(K_i)^D , is shared among the users in D_G D_i. In this case, each resource is distributed in a single multicast session, and the users may subscribe to one or several multicast sessions according to their access privilege. The task of key management is to securely update and distribute to the users in S_i , where i=1,2,3......m.

In the second method, the users in each SG share a secrete key called the Service Group Key and the multicast sessions are formed based on SGs. In particular, the users in share the service group key (K_i)^S and form one multicast session. In this multicast session, the resources are encrypted by (K_i)^S and transmitted to the users in Si . In this case, one resource may be distributed in several multicast sessions while being encrypted by different service group keys. The task of key management is to securely distribute and update(K_i)^S for the users in SG .The first session contains all users, and distributes resource r_1 encrypted by (K_1)^D . The second session contains users in S_1 and S_2, and distributes resource r_2 encrypted by(K_2)^D . The third session contains users in S_3 , and distributes resource r_3 encrypted by (K_3)^D . The communication overhead of a multicast session can be described as DR(r_i)G(x) , where DR(r_i) denotes the data rate of resource r_i , and G(x) is the cost of sending unit data to x users through multicast.

3.2 Member Key Generation

The GM initializes members' registration by generating a username for each intending member. Members later complete their registration process by individually login to the key server using their given usernames. At this stage, the KDC generates private/public key pair for each member. This is to prevent multiple usage of a single public/private key pair by different members and to eliminate the need for public key certificate issuance by the GM. The key generation algorithm is given in algorithm 1.

ALGORITHM 1:Key Generation Process

Let S= {M_1, $M_{2,}$………, M_n} denotes the set of all members in the service group S. The KDC generates public/private key pair for a member $M_i \in$ S (i=1, 2, 3,….., n) by following these steps:

Step 1: KDC chooses a private integer dmi for member Mi such that $dm_i \neq dm_j$.

Step 2: KDC calculates $eM_i(x_2, y_2) = dM_i \times G$

Step 3: KDC generates an individual prime number PM_i such that gcd(PM_i, PM_j) =1. This is M_i's CRT modulus.

Step 4: KDC then announces (eM_i, PM_i) as M_i's public key to all other members while (dM_i , PM_i) is known only to M_i (M_i's private key).

Finally, M_i's username U_{id} M_i and his public key (eM_i, PM_i) are registered and publicly announced as his public parameters. On the other hand, the private key is known to M_i and KDC..

3.3 Group Operation

In case of join operation in multi-level group access, when a user come then all the key's from the point of join to root all are updated and encrypted so that backward secrecy should be maintain. In case of multi-level group access control when a user come, then it provide its identity and service that it want to access. All operations are performing on figure 1 where three group are formed according to required service. Here S_{k1}, S_{k2} and S_{k3} are service group key while 1,2, 3……..11 and 12 keys are generated for users M_1, M_2………..M_{11} and M_{12} users. According to service group users create the intermediate keys such as for group 1 keys are k_{12}, k_{34}, K_1^s and K_1^D. In this paper an algorithm is given which used represents the procedure of a member join. This process expects the member identity (MEMID) and the identity key (k_d) of the member to be joined and results in a rekeying message. Tackle the complexity of this algorithm; it is divided into four coarse steps. The first two steps relate to tree management and depend on its mode, i.e., static or dynamic. Step 3 represents the actual rekeying including the generation of new keys, the encryption of new keys, and the estimation of the hash value of the rekeying message. The resulting hash value h is then digitally signed according to Algorithm 2.

ALGORITHM 2: Join Operation

```
Input: MEMID, $k_d$
Output: Rekeying message
  1. Add new leaf (MEMID, $k_d$)
  2. Update tree topology data (MEMID)
  3. Update keys on the join path (MEMID) -> Hash value (h)
  4. Sign hash value (h)
```

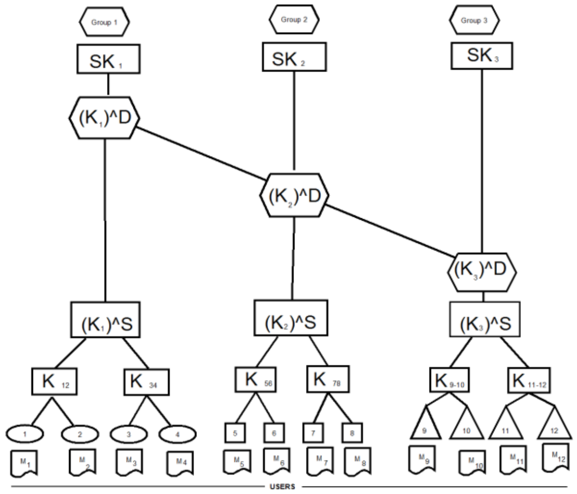

Fig. 1: Integrated Key Graph (Initial Group)

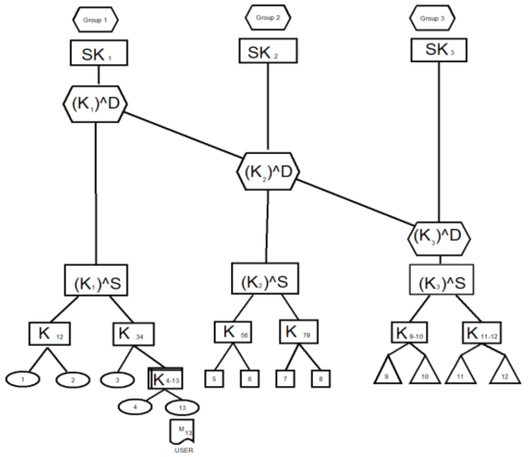

Fig. 2: Joining operation for user 13

As shown in the figure 2 and 3, the following keys are updated when a member M (k_{13}) is joining the group. New keys will be ($k_{4\text{-}13}$),($k_{3\text{-}4}$), (K_1)^S, (K_1)^D and SK_1. The following rekeying sub message are constructed Ek$_{13}$[(new k $_{4\text{-}13}$], E_4[new $k_{4\text{-}13}$], Enew $k_{4\text{-}13}$[new $_{k3\text{-}4]}$, Ek$_3$[new $k_{3\text{-}4}$], Enew $k_{3\text{-}4}$ [new (K_1)^S], Ek$_{1\text{-}2}$[new (K_1)^S] Enew (K_1)^S [new(K_1)^D] E(K_2)^D [new (K_1)^D] and Enew (K_1)^D[SK_1].

Multiple Joining Operations

Now when two or more member join in the different service group then all the keys from leaf's where the members are joins to session keys (root) are updated to provide backward secrecy . Now we show two user m_{13} and m_{14} joining case in the figure 3. User m_{13} join in the S_1 service group and m_{14} join in the S_2 service group. Now it nee d to update the following keys $k_{4\text{-}8}$, $k_{3\text{-}4}$, (K_1)^S, (K_1)^D, SK_1, $k_{8\text{-}14}$, $k_{7\text{-}8}$, (K_2)^S, (K_2)^D and SK_2. and following rekeying sub messages Ek$_{13}$[(new k $_{4\text{-}13}$], E_4[new $k_{4\text{-}13}$], Enew $k_{4\text{-}13}$[new $k_{3\text{-}4}$] , Ek$_3$[new $k_{3\text{-}4}$], Enew $k_{3\text{-}4}$ [new (K_1)^S], Ek$_{1\text{-}2}$[new (K_1)^S] Enew (K_1)^S [new(K_1)^D] E(K_2)^D [new (K_1)^D] Enew (K_1)^D[SK_1], Ek$_{14}$[$k_{8\text{-}14}$], Ek$_8$[$k_{8\text{-}14}$], Enewk$_{8\text{-}14}$[newk$_{7\text{-}8}$],Ek$_7$[new$_{7\text{-}8}$],Enewk$_{7\text{-}8}$[new(K_2)^S], and E$_{new}$$(K_2)$^S[$(K_2)$^D], Enew (K_2)^D[(K_1)^D] and E$_{new}$ (K_2)^D[SK_2]

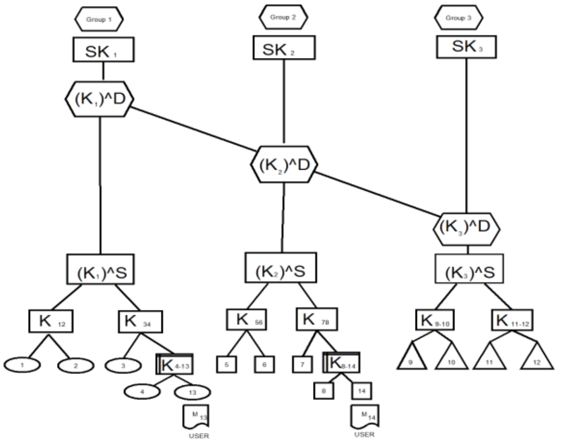

Fig .3: Multiple Joining for User 13 and 14

Leave Operation

In figure 4 user M_4 wants to leave the group then how many encryptions have to be computed by the server to rekey the group so that forward secrecy should be maintain .

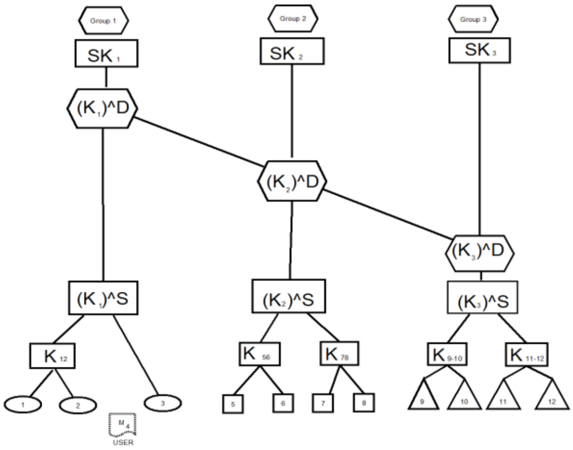

Fig.4: Leaving Operation for user 4

Except the identity key, all the keys held by M_4 (i.e.,$k_{3\text{-}4}$, (K_1)^S,(K_1)^D and SK_1) have to be changed. After removing M_4, however, the help-key k3-4 can be destroyed, as this key is only known by one member, i.e., m_3. Therefore, only three keys new (K_1)^S, new(K_1)^D and new SK_1 are generated, encrypted, and sent to the remaining members needing these keys. The server has to be generate the following rekeying sub message in order to disjoin the member M_4 from the multi-level group communication Ek$_3$[new(K_1)^S], Ek$_{1\text{-}2}$ [new (K_1)^S], Enew (K_1)^S[new(K_1)^D], E(K_2)^D[new (K_1)^D] and Enew (K_1)^D[new SK_1]

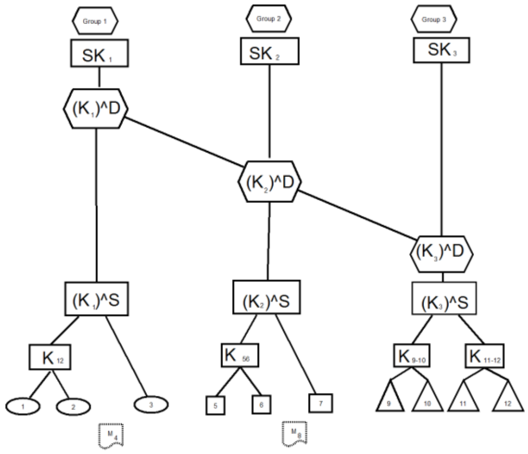

Fig.5: Multiple Leaving for User 4 and 8

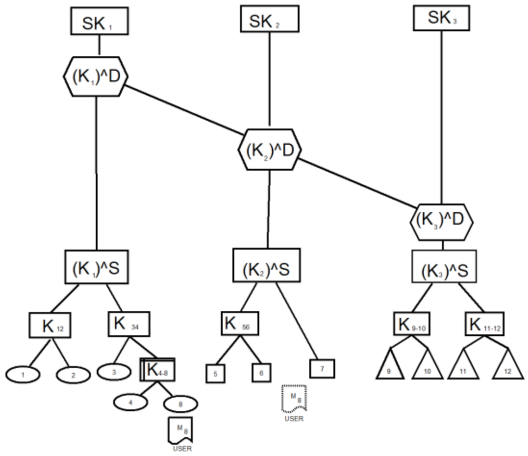

Fig.6: Switching of User 8 from S2 to S1

Multiple Leaving Operations

Now when two or more member are leave in the different service group then all keys from leaf's where the members are leaves to session keys (root) are updated to provide forward secrecy. Now, we show in figure 5 when member 4 and member 8 leave the service group from 1 and 2. Then following keys are updated to provide forward secrecy (K_1)^S, $(K_1$^D), SK_1, (K_2)^S, (K_2)^D and SK_2.

Switch Operation

Switch Operation is combination of join and leave operation in multi level group communication. The user switches from one service group to another service group.

Leave operation can take place when user switch from the current service group and maintain the forward secrecy and Join operation take place in the service group where the user join and maintain the backward secrecy in that service group.

In the figure 6 show that user M_8 switches from the service group S_2 to S_1. In service group S_1 we apply the leave operation and in service group S_2 we apply the join operation at a time. So the following keys are updated in case switch k_{4-8}, k_{3-4}, (K_1)^S, (K_1)^D, SK_1, (K_2)^S, (K_2)^D, SK_2. and the following rekeying sub messages are generated Ek_4[newk_{4-8}], Ek_8[new k_{4-8}] , Eknew$_{4-8}$[new k_{3-4}], Eknew$_3$[new k_{3-4}], Ek_{1-2}[new (K_1)^S], Enewk_{3-4}[(K_1)^S], Enew (K_1)^S[(K_1)^D], Enew(K_1)^D[newSK_1], Ek_7[new(K_2)^S], Ek_{5-6}[new (K_2)^S], Enew(K_2)^S[new (K_2)^D], E(K_3)^S[new (K_2)^D], Enew(K_2)^D[new(K_1)^D] and Enew (K_2)^D[new SK_2].

4. PERFORMANCE EVOLUTION

Here, the scheme can be compared with some existing secure Group Key management protocols that deal with establishing a session key among several users for secure group communications. Number of multiplications and signature verifications are determined in terms of total number t of participating members in a session. It defined the performance measures as

Storage overhead at the KDC: it is defined as the expected number of keys stored at the KDC.

Rekey overhead at the KDC: it is defined as the expected amount of rekeying message transmitted by the KDC.

Storage overhead of users: it is defined as the expected number of keys stored by the users in service group.

Rekey overhead of users: It is defined as the expected amount of rekeying messages received by users in service group.

Communication Overhead: communication overhead is defined in terms of bandwidth utilization, delay and key size.

Now, it can observe on the base of theoretical concept that in case of ECC all the key size is near 160 bit that provides same level of security as we use Diffie-Hellman because in case ECC point multiplication is very difficult process. So this property provides the same level of security. So by using ECC the key message size should be decrease and so communication. Overhead should be decrease and the storage overhead also decrease in the multi-level secure group communication. The details of comparison of performance parameter with graphs are illustrated in the extension of this paper.

5. CONCLUSION

This paper is providing confidentiality and authenticity for messages exchanged between members of a particular group is an important issue. In this paper, on the base of theoretical observation a dynamic multicast encryption scheme for secure multi-level group communication using the elliptic curve cryptography. The scheme is secure, simple and easy to realize. It also achieves efficiency in both computational and communication aspects while enhancing constant size cipher text, stateless receivers, confidentiality, integrity, low bandwidth and storage consumption as well as perfect forward and backward secrecy. With these numerous advantages, we believe that the approach can be extended to social network scenarios for disseminating highly classified information among several users

6. REFERENCES

[1] Wenbo Sh and Peng Gong, “A New User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography”, International Journal of Distributed Sensor Networks Volume 2013 (2013), PP 1- 7.

[2] H. Debiao, C. Jianhua, and H. Jin, “An Id-Based Client Authentication With Key Agreement Protocol For Mobile Client-Server Environment On Ecc With Provable Security,” Information Fusion, vol. 13, no. 3, pp. 223–230, 2012.

[3] Guiyi Wei, Xianbo Yang and Jun Shao, “Efficient Certificateless Authenticated Asymmetric Group Key Agreement Protocol,” KSII Transactions On Internet And Information Systems, Vol. 6, No.12, pp.3352-3365, Dec 2012.

[4] Sandeep S. Kulkarni, Bezawada Bruhadeshwar, “Key-update distribution in secure group communication,” Elsevier, Computer Communications,Vol. 33,No.6, pp.689-705, 2010.

[5] Yan , Sun "Hierarchical Group Access Control for Secure Multicast Communication " IEEE 2007

[6] Y. Mao, Y. Sun, M. Wu, and K. J. R. Liu, “JET: Dynamic joint-exist-tree amortization and scheduling for contributory key agreement,” IEEE/ACM Trans. Netw., vol. 14, no. 5, pp. 1128–1140, Oct. 2006.

[7] W. Trappe, Y.Wang, and K. J. R. Liu, “Resource-aware conference key establishment for heterogeneous networks,” IEEE/ACM Trans. Netw., vol. 13, no. 1, pp. 134–146, Feb. 2005.

[8] Y. Sun and K. J. R. Liu, “Scalable Hierarchical Access Control in Secure Group Communications,” in Proc. IEEE INFOCOM, Mar. 2004.

[9] Y. Sun, W. Trappe, and K. J. R. Liu, “A scalable multicast key management scheme for heterogeneous wireless networks,” IEEE/ACM Trans. Netw., vol. 12, no. 4, pp. 653–666, Aug. 2004.

[10] B. Sun, W. Trappe, Y. Sun, and K. J. R. Liu, “A time-efficient contributory key agreement scheme for secure group communications,” in Proc. IEEE Int. Conf. Commun. (ICC), 2002, vol. 2, pp. 1159–1163.

[11] A. Perrig and J. D. Tygar, Secure Broadcast Communication: In Wired and Wireless Networks. Norwell, MA: Kluwer, 2002.

[12] S. Banerjee and B. Bhattacharjee, “Scalable secure group communication over IP multicast,” IEEE J. Sel. Areas Commun., vol. 20, no. 8, pp. 1511–1527, Oct. 2002.

[13] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam, “Reliable group rekeying: A performance analysis,” in Proc. 2001 Conf. Applications, Technologies, Architectures, Protocols For Comput. Commun., Aug. 2001, pp. 27–38.

[14] A. Perrig, D. Song, and D. Tygar, “ELK, a new protocol for efficient large-group key distribution,” in Proc. IEEE Symp. Security Privacy, 2001, pp. 247–262.

[15] W. Trappe, J. Song, R. Poovendran, and K. J. R. Liu, “Key distribution for secure multimedia multicasts via data embedding,” in Proc. IEEE ICASSP’01, May 2001, p. 1449–1452.