Computing Efficient Rekeying for Multicast Key Distribution with Least Computation Complexity

Nilesh M.Shidurkar
Asst.Professor(cse)
S.E.C. Washim
India

ABSTRACT
An imperative issue for secure gathering openness is of the utmost importance dissemination. The majority of the incorporated gathering key administration plans utilize high rekeying expense. Here we present a novel methodology for calculation productive rekeying for multicast key conveyance. This methodology diminishes the rekeying expense by utilizing a mixture gathering key administration plan (including both concentrated and contributory key administration plans). The gathering controller uses the MDS Codes, a class of blunder control codes, to circulate the multicast key powerfully. Keeping in mind the end goal to maintain a strategic distance from successive rekeying as and when the client leaves, a novel methodology is presented where customers recompute the new gathering key with negligible calculation. This methodology guarantees forward mystery and additionally in reverse mystery and fundamentally lessens the rekeying expense and correspondence cost. This plan well suits remote applications where compact gadgets require low calculation.

Keywords
Erasure decoding, Key Distribution, MDSCodes, Multicast.

1. INTRODUCTION
Security is vital for information transmission through an unstable system. There are a few plans to address the unicast security issues however they can't be straightforwardly reached out to a multicast situation. By and large, multicasting is significantly more powerless [4, 5, 6] than unicast in light of the fact that the transmission happens over numerous system channels. A more troublesome and testing issue emerges because of the multicast bunch enrollment being rapid. Clients can leave and join the gatherings, accordingly making the issue of gathering administration more troublesome in huge scale frameworks. Additionally we have to give Forward Secrecy and Backward Secrecy.One of the most critical issues in Multicast Security is the Group Key Management. Gathering key administration, which is worried with producing and upgrading mystery keys, is one of the essential advances to secure such gathering interchanges. Key management facilitates access control and data confidentiality by ensuring that the keys used to encrypt group communication are shared only among legitimate group members. Thus, only legitimate group members can access group communications. The shared group key can also be used for authentication. When a message is encrypted using the group key, the message must be from a legitimate group member. To prevent these problems, the following two security criteria are important for the group key distribution in secure multicast communication. Forward secrecy: If a person has left a group, the departed member cannot decrypt encrypted messages transmitted after the leaving. Backward secrecy: If a person joins a group, he cannot decrypt encrypted messages transmitted before the joining. The process for achieving forward and backward secrecy requires redistributing the group key. This process is called group rekeying [7][13].

With the growth of the Internet, the usage of group communication becomes more popular. These applications include the pay TV channels, secure video conferencing, multi-partner military action, wireless sensor, and ad hoc networks. In today's era, information security is the prime concern as with the technological advancements, the attackers are provided with more powerful and sophisticated tools. Today, the Internet is not totally secure for privacy. The usage of multicast applications increases day by day so it needs secure multicast services. Multicasting is a simple way to send one stream of data to multiple users simultaneously. It helps in reducing the required bandwidth significantly, as it enables splitting of a single transmission between multiple users [9]. Multicasting not only optimizes the performance, but also enhances the efficiency of the network. For these reasons, multicasting has become the preferred transmission method for most group communication.

Gathering key administration assumes a vital part in gathering correspondence. A typical gathering key is required for individual clients in the gathering for secure multicast correspondence. Gathering key must be overhauled as often as possible at whatever point a part joins and leaves keeping in mind the end goal to give forward and in reverse mystery. Forward mystery guarantees that a removed part can't assemble data about future multicast correspondence and in reverse mystery guarantees that a joining part can't accumulate data about past multicast correspondence [11]. For this reason, group key needs to be updated with each membership change and given away to the authenticated users. This process is known as group re-keying. For group communication , Wong et al. and Waller et al. has proposed a scheme 'logical key hierarchy (LKH) tree approach' [3, 4] which provides an efficient and secure mechanism to maintain the keys. In addition, communication and computation cost increases logarithmically with the group size for a join or depart request. Communication cost in LKH is reduced from O(n) to O(logn) in the rekeying method, where n is the number of group members. One-way function (OFT) scheme was proposed by Sherman and McGrew [5] to reduce the communication cost from 2logn−1 to logn.These schemes need to rekeying message whenever member joins/leaves the group.This eliminates the need to unicast the secret keys to every member separately, henceforth, reducing the load on the server to great extent. The proposed scheme differs from the previous work as it is always maintaining the forward and backward secrecy. In addition, our scheme does not need to maintain the key tree topology and eliminates the rekeying

process whenever member joins/leaves the group. In the aspect of security, our proposed scheme guarantees the group secrecy, forward secrecy, backward secrecy.There are three types of group key management schemes. In centralized key management, such as, group members trust a centralized server, referred to as the key distribution center (KDC), which generates and distributes encryption keys. In decentralized schemes, the task of KDC is divided among subgroup managers. In contributory key management schemes, group members are trusted equally and all participate in key establishment [8][12][14].

In this paper, we study how a multicast group key can efficiently be distributed in computation. In this a centralized key management model is used where session keys are issued and distributed by a central group controller (GC), as it has much less communication complexity, when compared to distributed key exchange protocols. The group controller uses the communication,Computation and storage resources for distributing the session key to the group members. The main problem here is how the resources can be used to distribute the session key. This is referred to as group key distribution problem. There are two approaches that are generally used for distributing the session key to the group of n members. The first approach is that the group controller GC shares an individual key with each group member. That key is used to encrypt a new group session key. In the second approach the group controller shares an individual key with each subset of the group, which can then be used to multicast a session key to a designated subset of group members. This approach has less communication, computation and storage complexity when compared to the other approach.

For a multicast group with large number of members key-tree-based approach is used. This approach decomposes a large group into multiple layers of subgroups with smaller sizes. Using this approach communication complexity is reduced, but the storage and computation complexity is increased. A new novel approach for computation efficient rekeying for multicast key distribution is introduced. This approach reduces the rekeying cost by employing a hybrid group key management scheme and also maintains the same security level without increasing the communication and storage complexity. In this scheme, session keys are encoded using error control codes. In general encoding and decoding using error control code reduces the computation complexity. Thus, the computational complexity of key distribution can be significantly reduced.

2. LITERATURE SURVEY

2.1 Efficient Computation for Distribution

An important problem for secure group communication is key distribution. In this paper, a new multicast key distribution scheme[10] is introduced whose computation cost is significantly reduced. This scheme employs MDS Codes, a class of error control codes, to distribute multicast key dynamically. This reduces the computation load of each group member. When this scheme is used with key-tree-based schemes, it provides much lower computation complexity which also maintains low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution.

In key distribution scheme, a basic operation is to distribute a piece of secret data to a small group of n members, where each shares a different key with the GC. In the existing schemes, this is done by n encryptions, followed by n unicasts. In the new scheme, this is done by using one erasure decoding of certain MDS code, followed by one multicast to all n members. This is the basic key distribution scheme of key distribution that is used in this paper. This scheme can be integrated into any key distribution scheme, especially the schemes based on key trees, to reduce the computation cost. The multicast group that is used can have n members.

2.2 Iolus Approach

Iolus approach [2] proposed the notion of hierarchy subgroup for scalable and secure mulitcast.In this method, a large communication group is divided into smaller subgroups. Each subgroup is treated almost like a separate multicast group and is managed by a trusted group security intermediary (GSI). GSI connect between the subgroups and share the subgroup key with each of their subgroup members. GSIs act as message relays and key translators between the subgroups by receiving the multicast messages from one subgroup, decrypting them and then remulticasting them to the next subgroup after encrypting them by the subgroup key of the next subgroup. The GSIs are also grouped in a top-level group that is managed by a group security controller (GSC). Although Iolus has improved the scalability of the system, because the member join or leave only affect their subgroup only while the other subgroup will not be affected. It has the drawback of affecting data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup, and thereby one key, to another. This becomes even more problematic when it takes into account that the GSI has to manage the subgroup and perform the translation needed. The GSI may thus becomes the bottleneck.

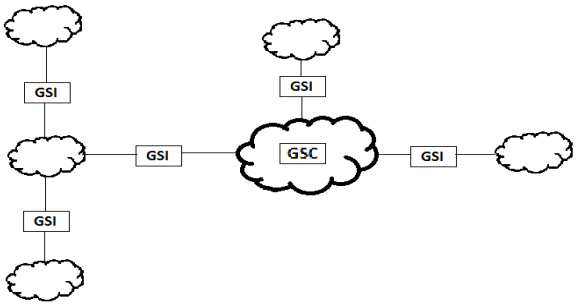

Fig 1 : Secure Distribution Tree.

2.3 Logical Tree Structure

The logical key hierarchy (LKH) [11] is an efficient approach that supports dynamic group membership. This method was proposed by Wallner et al. and Wong et al. individually. Waller et al. discussed binary trees and Wong et al. discussed the generalized case - key graphs, but the implicated ideas in their method is identical – to convert the cost of communication from linearly to logarithm with the group size of n. In this approach, the group controller (GC) maintains a logical key tree where each node represents a key encryption key (KEK). The root of the key tree is the group key used for encrypting data in group communications and it is shared by all users. The leave node of the key tree is associated with a user in the communication group. Each user secretly maintains the keys related to the nodes in the path from its leaf node to the root. We call the set of keys that a member knows the key path. Figure 1 shows a sample of key tree. When a member leaves the group, all the keys that the member knows, including the group key and its key path, need to be refreshed. When a member joins the group, GC authenticates the member and assigns it to a leaf node of the

key tree. The GC will send the new member all the keys from his/her corresponding leaf node to the root. The main reason for using such a key tree is to efficiently update the group key if a member joins or leaves the group.

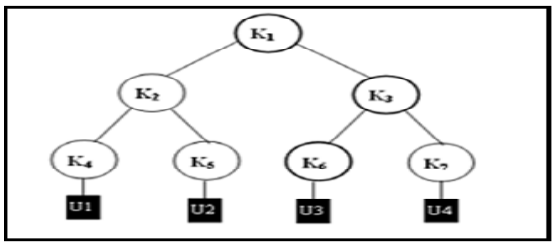

Fig 2 : Logical Tree Structure.

3. EXISTING METHODOLOGIES AND ANALYSIS

For an element multicast bunch, a session key is issued by Group Controller (GC). The Group Controller uses this session key to build up a protected multicast with the approved gathering individuals. At the point when new individuals join or leave the gathering, the GC reissues the new session key to the confirmed gathering individuals. This guarantees security of the present session and that of the old sessions. That is the recently joined individuals can't recoup the interchanges can not get to the present session. Therefore the forward mystery and in reverse mystery is kept up for the gathering correspondence is kept up.

The complexity of the rekeying operation changes when new members join the group and old members leave the group. When a new member join the group, the GC multicast the new session key encrypted by the current session key to all the current members, followed by a unicast to the new member to send the new session key encrypted by a predetermined encryption key shared between the GC and the new member. Thus, with low computation cost and communication cost a new member can join the group. However, when an old member leaves the group , the current session key cannot be used to convey the new session key securely, since it is known to the old member.

In key appropriation conspire, a fundamental operation is to circulate a bit of mystery information to a little gathering of n individuals, where every shares an alternate key with the GC. In the current plans, this is finished by n encryptions, trailed by n unicasts. In the new plan, this is finished by utilizing one deletion interpreting of certain MDS code, trailed by one multicast to all n individuals. This is the essential key circulation plan of key conveyance. This plan is incorporated into any key appropriation plan, particularly the plans taking into account key trees, to lessen the calculation cost. The multicast bunch that is utilized can have n individuals.

3.1 Maximum Distance Separable Codes Algorithm

It mainly consist of three parts, they are as follows:

a) Initializing Group controller.

b) Subscribing new members.

c) Applying the procedure of Re-Keying whenever member leaves the group.

Steps for the Algorithm:

Step I : GC Initialization by constructing codeword C using MDS.

Step II : Applying One-Way Hashfunction

Step III : H(x)=y, property of Hashfunction

Step IV: Subscribing new member

Step V: Ji= +ve integer Ji!=Jk

Step VI : Select Si

Step VII : Applying the procedure of Re-Keying whenever member leaves the group.

Step VIII: Cj=H(Si+r)

Step IX: Member j every ‘n’ members in the group calculates these own codeword C1,C2,……,Cn

4. PROPOSED WORK

(i) Join operation

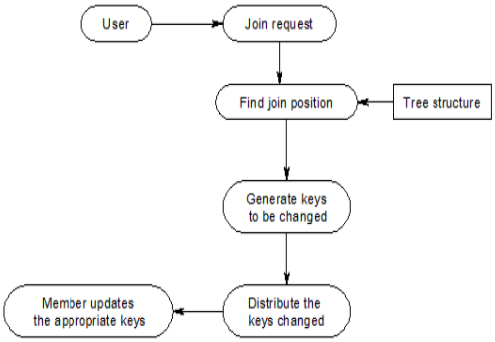

Fig 3: Diagram for Join Operation

(ii) Leave Operation

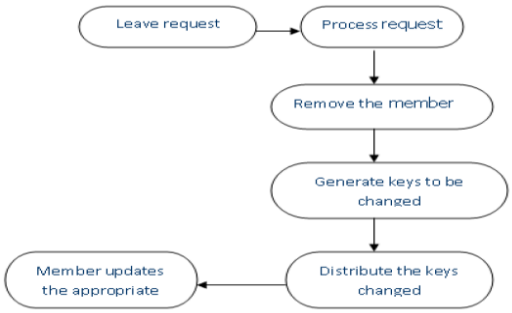

Fig 4: Diagram for Leave Operation

4.1 Novel approach for Computation-Efficient Rekeying

A set of dummy user are introduced by the server inorder to protect the size of the group (which plays a critical role in our approach). The dummy users introduced by the server randomly join or leaves the group. Now at anypoint of time the members in the group will be as GrpSizeold= uj +dj – (ul + dl), where uj and ul is user join and user leave and d j and d l is dummy user join and dummy user leave. Inorder to protect the group key information even when a user leaves, we consider the group size as the critical factor. It is understood that in group communication member join and member leave is a dynamic process. When a member leaves the group key should be redistributed and so computation cost becomes more tedious.

To calculate the new group key, the authenticated group member executes the following steps:

1. Initially, the GC computes the group key GrpKey and distributes to users by using the MDS Codes[10].
2. When uj no of user leaves the group, server randomly introduces djnew and dlleave. The user uj who left the group cannot predict the group size changes that has made in the group after he leaves.
3. Now the group size will be GrpSizenew= GrpSizeold + uin + din – (uout + dout) where uin is the no of members joining the group, uout is the no of members leaving the group, din is the no of dummy users joining the group and dout is the no of dummy users leaving the group.
4. The new group key is calculated as GrpKeynew = GrpSizenew + GrpKey.
5. Now a new value j is calculated such that j= GrpSizenew mod 64.
6. The new group key GrpKeynew is updated by undergoing a cyclic shift of GrpKeynew.

The steps 2,3,4,5,6 continues when the user leaves the group. Thus a new group key is calculated by each group members and rekeying is done This makes the computation cost less and the rekeying is more significant. But, in the earlier approach the computation cost is more because the multicasting is done at every rekeying process.

For security reasons, the rekeying using MDS codes has to be done in some interval. The frequency of rekeying is much lesser than earlier case when rekeying is done for every user leave. This subsequently reduces the rekeying cost and significantly improves the security.

Moreover the group dynamic membership information such as group size ,number of user joining, number of user leaving is unknown to any user.

5. IMPLEMENTATION DETAILS

Implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, and running, testing, and making necessary changes. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. Following models helps to get precise model of our paper:

Fig 5: Node 6 sending leave request to Cluster head1 (CH1)

Fig 6: Re-keying done by Cluster head1 (CH1)

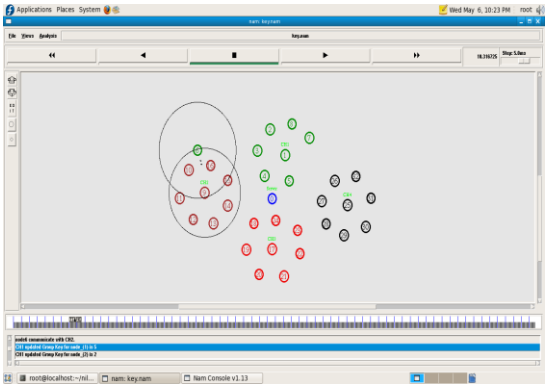

Fig 7: Node 6 sending Join request to Cluster head2 (CH2)

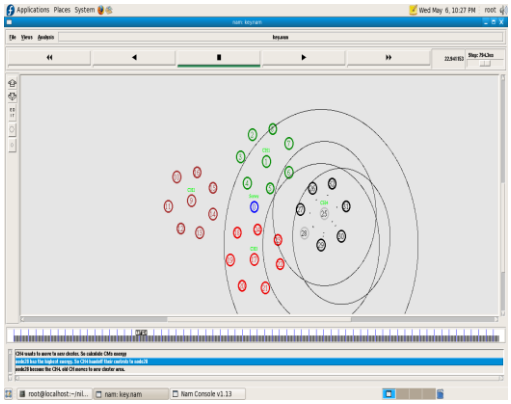

Fig 8: Calculating highest energy of node in Cluster head3 (CH3).

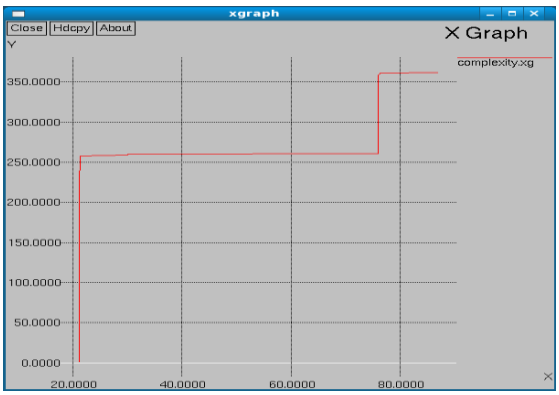

Fig 9: Communication Complexity graph.

6. CONCLUSION

The objective is to provide the security along with the least computation for joining and leave operation of participants.

In this paper a new approach is used which makes the computation cost much more efficient and the rekeying cost is significantly reduced. The group key is multicasted by the GC to the group members using the MDS Codes. Frequent rekeying is avoided when the user leaves, where clients recompute the new group key with minimal computation. This also makes the computation complexity greatly reduced.

7. REFERENCES

[1] X.S. Li, Y.R. Yang, M.G. Gouda, and S.S. Lam, "Batch Rekeying for Secure Group Communications," Proc. 10th Int'l World Wide Web Conf. (WWW '01), pp. 525-534, May 2001.

[2] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting," Proc. ACM SIGCOMM '97, pp. 277-288, Sept. 1997.

[3] C. Wong, M. Gouda, and S. Lam, "Secure Group Communication Using KeyGraphs," IEEE/ACM Trans. Networking, vol. 8, pp.12-23, Feb. 2000.

[4] Peter S. Kruus and Joseph P. Macker, "Techniques and issues in multicast security," MILCOM98, 1998.

[5] Paul Judge and Mostafa Ammar**,** "Security Issues and Solutions in Multicast Content Distribution: A Survey", IEEE Network, February 2003, pp 30 – 36.

[6] M. Moyer, J. Rao and P. Rohatgi, "A Survey of Security Issues in Multicast Communications", IEEE Network Magazine, Vol. 13, No.6, March 1999, pp. 12-23.

[7] Yan Sun, and K.J. Ray Liu, "Securing Dynamic Membership Information in Multicast Communications," IEEE INFO CONFERENCE 2004.

[8] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error Correcting Codes. North- Holland Math. Library, 1977.

[9] M. S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications," IEEE Trans. Veh. Technol., vol. 48, pp. 1469–1474, Sept. 1999.

[10] Lihao Xu, Cheng Huang, "Computation Efficient Multicast Key Distribution," IEEE Trans. Parallel And Distributed Systems, Vol 19, No. 5, May 2008.

[11] Ran Canatti, Juan Garay, Gene Itkis, Daniel Micciancio, Moni Naor, and Benny Pankus, "Multicast Security: A taxanomy and some efficient constructions", IEEE Network, March 1999, pp 122-128.

[12] H. Harney and E. Harder, Logical Key Hierarchy Protocol, IETF Internet draft, work in progress, Mar. 1999.

[13] T.M. Cover and J.A. Thomas, Elements of Information Theory. John Wiley & Sons, 1991.

[14] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, fourth ed. CRC Press, 1999.