International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014
ISSN 2091-2730

A Survey on Rekeying Framework for Secure Multicast

HITESH PATIL M.TECH STUDENT AURNAGABAD(MS) INDIA

hiteshppatil15@gmail.com MOB.NO. +919028173530

Abstract— Group key management (GKM) in mobile communication is important to enable access control for a group of users. A major issue in GKM is how to minimize the communication cost for group rekeying. To design the optimal GKM, researchers have assumed that all group members have the same leaving probabilities and that the tree is balanced and complete to simplify analysis. In the real mobile computing environment, however, these assumptions are impractical and may lead to a large gap between the impractical analysis and the measurement in real-life situations, thus allowing for GKM schemes to incorporate only a specific number of users.

In this paper, we propose a new GKM framework supporting more general cases that do not require these assumptions. Our framework consists of two algorithms: one for initial construction of a basic key-tree and another for optimizing the key-tree after membership changes. The first algorithm enables the framework to generate an optimal key-tree that reflects the characteristics of users' leaving probabilities, and the second algorithm allows continual maintenance of communication with less overhead in group rekeying. Through simulations, we show that our GKM framework outperforms the previous one which is known to be the best balanced and complete structure.

.

Keywords— multicast, security, group key, group key management, logical key hierarchy, batch rekeying, group dynamics

1.INTRODUCTION.

Multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission. Such applications need a secure group key to communicate their data. This brings importance to key distribution techniques. For group-oriented applications, multicast is an essential mechanism to achieve scalable information distribution. Multicast describes communication where information is sent from one or more parties to a set of other parties. In this case, information is distributed from one or more senders to a set of receivers, but not to all users of the group. The advantage of multicast is that, it enables the desired applications to service many users without overloading a network and resources in the server.

Security is essential for data transmission through an insecure network. There are several schemes to address the unicast security issues but they cannot be directly extended to a multicast environment. In general, multicasting is far more vulnerable [1], [2], [3] than unicast because the transmission takes place over multiple network channels. In multicast group communication, all the authorized members share a session key, which will be changed dynamically to ensure forward and backward secrecy referred as "group rekeying".

2. COMPARISON OF GROUP KEY MANAGEMENT PROTOCOLS.

2.1 Centralized key management protocols: A single entity is employed for controlling the whole group; hence a group key management protocol seeks to minimize storage requirements, computational power on both client and server sides, and bandwidth utilization. Although the centralized approach has a problem of a single point of failure, some applications like stock quotes are still centralized. To overcome this problem, a mirror delivered to all group members. This causes the bandwidth waste because rekeying messages are delivered to members who do not need them as well as intended receivers.

2.2 Decentralized key management protocols: The management of a large group is divided among subgroup managers, trying to minimize the problem of concentrating the work in a single manager. These protocols need more trusted nodes and suffer from encryptions and decryptions processes between subgroup managers. Some examples of decentralized protocols are: Scalable Multicast Key Distribution using Core Based Tree (CBT) [15], Iolus [16], Dual-Encryption Protocol (DEP) [17] and Kronos [18]. Cheng and Laih [26] modified Tseng's conference key agreement protocol based on bilinear pairing. In 2009, Huang et al. [27] proposed a non-interactive protocol based on DL assumption to improve the efficiency of Tseng's protocol.

2.3 Distributed key management protocols: There is no explicit manager, and the members themselves do the key generation. All members can perform access control and the generation of the key can be rather contributory, meaning that all members contribute some information to generate the group key, or done by one of the members. The distributed protocols have a scalability problem in case of key update, since they require performing large computations and they are characterized by large communication overheads. Further, they need all group members to have powerful resources. Some examples of distributed key management protocols are: Octopus Protocol [2], Distributed Logical Key Hierarchy [15] and Diffie-Hellman Logical Key Hierarchy [8]. In the following subsection, an overview of the proposed protocol is given.

For secure multicast services, various tree based group key management schemes have been introduced until now. Traditional tree based approaches uses conventional encryption algorithms which focus on reducing the number of rekeying messages transmitted by the key distribution center (group manager/controller). However, they do not consider the network bandwidth used for transmitting each rekeying message. To provide a scalable rekeying, the key tree approach makes use of KEKs so that the rekeying cost increases logarithmically with the group size for a join or depart request An individual key serves the same function as KEK, except that it is shared only by the GC and an individual member [21]. To this end, KDC aggregates multiple rekeying messages into one multicast flow, which is referred to as group oriented rekeying [7]. In group oriented rekeying, all rekeying messages are members. This operation [12] is done by GC using one erasure decoding of certain MDS code, followed by one multicast to all the n members In this approach, the rekeying is done at every member join or leave. The new group key is multicasted to the group members each time by the group controller through multicasting to establish security. In this scheme, the GC has to communicate with the group members each time. The complexity of the rekeying operation changes because rekeying is done at every member joins or leaves the group, which results in high computational complexity. In this scheme , when a member leaves the group, rekeying operation is performed to compute the new group key, which increases the burden on the server to recompute the group key and then multicast to all the members of the group. Since it is dynamic in nature, several rekeying operation is taking place.

3. A New Secure Multicast Key Distribution Protocol Using Combinatorial Boolean Approach .

The proposed protocol in this scheme is based on Key Management using Boolean Function Minimization (KM-BFM) technique [11]. KM-BFM protocol is considered an enhancement to LKH protocols. Instead of using one tree as in KM-BFM; the members are divided into a number of subgroup trees. The group manager holds n key pairs and each group member holds y keys. The proposed protocol achieves a lower storage at both the group manager and the group members compared to KM-BFM protocol. It has to be noted that the authentication problem is not addressed in the present paper.

This protocol achieves a lower storage at both the group manager and the group members compared to KM-BFM protocol. Also, it has a lower update message length in case of a single member leave and a comparable update message length in case of multiple leaves. Furthermore, the probability of conducting a successful collusion attack in the proposed protocol is less than that proposed in KM-BFM protocol.

4. A new probabilistic rekeying method for secure multicast groups .

The Probabilistic optimization of LKH (PLKH) scheme [19], optimized rekey cost by organizing LKH tree with user rekey characteristic. This paper concentrate on further reducing the rekey cost by organizing LKH tree with respect to compromise probabilities of members using new join and leave operations.

The key identifier assignment requires more memory to store key identifiers. Though total nodes created are less than PLKH & LKH schemes, this scheme treats some nodes harshly in terms of depth assigned. Finally, this scheme only ensures that tree structure is binary. It neither tries to maintain strict binary tree as PLKH nor tries to balance all nodes at same level as LKH.

5. Optimal Communication Complexity of Generic Multicast Key Distribution

This scheme deals with tight lower bound on the communication complexity of secure multicast key distribution protocols in which rekey messages are built using symmetric-key encryption, pseudorandom generators and secret sharing schemes [22]. Updating the group key for each group membership change is at least $\log_2(n) - O(1)$ basic rekey messages. Key distribution in multicast is implemented using a central distribution authority, called the group center, responsible for establishing a shared key among all privileged group members, and for “rekeying” the group every time a new member joins and/or an existing member leaves the group. This lower bound involves defining a sequence of adversarial-chosen REPLACE operations (simultaneous execution of a LEAVE and JOIN) and every protocol incurs an average communication cost of $\log_2 n$ for such a sequence and every individual LEAVE performed for a cost of $\log_2(n)$ multicast messages and every individual JOIN for $\log_2(n)$ unicast messages.

6.Rekeying using MDS code on PFMH tree

The PFMH tree follows a PACK protocol, in which each group member equally contributes its share to the group key, and this share is never relieved to the others. PACK includes a set of rekeying protocols to update the group key upon group

membership change events for security purpose. The PACK protocol can achieve the minimum rekeying time cost upon membership change events. For any single-user Join event, the rekeying cost is O(1), and for any single user leave event, the rekeying time cost is of O(log n). The communication and computation costs can still be reduced by adopting PFMH tree and by introducing phantom nodes in the key tree.

In this scheme, each member will maintain and update the global key tree locally. Each group member knows all the subgroup keys on its key path and knows the ID and the exact location of any other current group member in the key tree. In PACK, when a new user joins the group, it will always be attached to the root of the join tree to achieve O(1) rekeying cost in terms of computation per user, time, and communication. When a user leaves the current group, according to the leaving member's location in the key tree, as well as whether this member has a phantom location in the key tree, different procedures will be applied, and the basic idea is to update the group key in O(log n) rounds and simultaneously reduce the communication and computation costs.

TABLE 1: SAMPLE TABLE COMPARISON OF KEY RECOVERY TIME

.

Multicast group size	PFMH Tree based key Distribution	Group Controller based key distribution
2	15.2 m. sec	27.6 m.sec
4	18.5m.sec	31 m.sec
6	21.5m.sec	35.8 m.sec
8	25.7m.sec	43.3 m.sec
10	27.5m.sec	46.6 m.sec

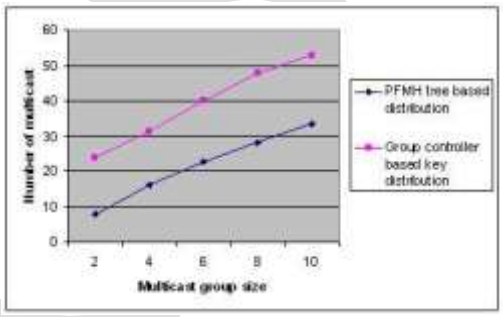

Fig: Computation cost

3.MY PROPOSED SYSTEM BASED ON SURVEY

To develop an Efficient Rekeying Framework for Secure Multicast with Diverse-Subscription-Period Mobile Users based on optimal GKM with dynamic mobile subscribers. The proposed framework consists of cost-efficient key-tree generation and management. We also provide a new mathematical analysis methodology for quantifying the performance of key-trees.
This frame work consider the following assumptions.

1. We propose a new mathematical analysis method-ology that can provide the precise average value of communication overhead for group key up-dates under general conditions. The conditions in-clude an arbitrary number of members, non-equal leaving probabilities, and non-balanced and non-complete tree structure can support the mobile situations. Note that unlike previous works, the average size of rekeying messages can be calculated even though the subscription periods are diverse. Also, through our analysis, we find the conditions for the optimal tree structure that minimizes com-munication overhead.
2. We develop a two-step mechanism for optimal key-tree generation: one for initial key-tree generation followed by key-tree maintenance after the group membership changes. The first algorithm can gen-erate a key-tree that corresponds to the optimal key-tree obtained by mathematical analysis.
3. For the second step of the mechanism in 2), we pro-pose an optimal key-tree maintenance algorithm for use after the group membership changes. The algorithm optimizes the key-tree by modifying the tree structure considering the diverse-subscription periods of the mobile users.

4.Acknowledgment

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendy advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I EXPRESS MY WARM THANKS TO PROFESSOR K.V.BHOSALE FOR THEIR SUPPORT AND GUIDANCE .

5.CONCLUSION

Key transfer protocols rely on a mutually trusted key generation center (KGC) to select session keys and transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration. We have optimized dynamic multicast key distribution scheme with MDS codes using PFMH tree. The computation complexity of key distribution is greatly reduced by employing erasure decoding of MDS codes instead of more expensive encryption and decryption computations.

REFERENCES:

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

MILCOM98,1998.

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

[3] 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.

[4] M. Waldvogel, G. Caronni, D. Sun, N. Weiler and B. Plattner, "The VersaKey Framework: Versatile Group Key Management", IEEE Journal on Selected Areas in Communications, 7(8), 1614-1631, August 1999.

[5] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting", Proc. of ACM SIGCOMMi'97, 277-288, Sep. 1997.

6. D. M. Wallner, E. J. Harder and R. C. Agee, "Key Management for Multicast: Issues and Architectures", Internet Draft (work in progress), draft-wallner-key-arch-01.txt, Sep. 15, 1998.

7. C. K. Wong, M. Gouda and S. S. Lam, "Secure Group Communications Using Key Graphs", Proc.ACM SIGCOMM'98, Sep. 1998.

8. Y. Kim, A. Perrig, and G. Tsudik, "Tree-Based Group Key Agreement," ACM Trans. Information and System Security, vol. 7, no. 1, pp. 60-96, Feb.

[9] S. Benson Edwin Raj, J. Jeffneil Lalith , "A Novel Approach for Computation-Efficient Rekeying for Multicast Key Distribution" IJCSNS , VOL.9 No.3, March 2009.

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

[11] Mohamed M. Nasreldin Rasslan, Yasser H. Dakroury, and
Heba K. Aslan "A New Secure Multicast Key Distribution Protocol Using Combinatorial Boolean Approach" ,International Journal of Network Security, Vol.8, No.1, PP.75–89, Jan. 2009

[12] C.Wong, M. Gouda, and S. Lam, "secure group Communications using key graphs," Proceedings of ACM SIGCOMM, pp. 68-79, Vancouver, British Columbia, September 1998.

[13] D. McGrew, and A. Sherman, Key Establishment in Large Dynamic Groups Using One-Way Function Trees, Technical Report No. 0755, TIS Labs at Network Associates, Inc., Glenwood, MD, May 1998.

[14] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D.
Saha,"Key management for secure internet multicast using Boolean function minimization techniques," Proceedings of the IEEE INFOCOM, vol. 2, pp. 689-698, New York, Mar. 1999.

[15] A. Ballardie, Scalable Multicast Key Distribution, RFC 1949, 1996.

[16] S. Mittra, "Iolus: A framework for scalable secure multicasting," Proceedings of the ACM SIGCOMM, vol. 27, no. 4, pp. 277-288, New York, Sep. 1997.

[17] L. Dondeti, S.Mukherjee and A. Samal, "Scalable secure
one-to-many group communication using dual encryption," IComputer and Communication, vol. 23, no. 17, pp. 1681-1701, Nov. 1999.

[18] S. Setia, S. Zhu, and S. Jajodia,"Kronos: A scalable group re-keying approach for secure multicast," Proceeding of the IEEE Symposium on Security and Praivcy, pp. 215-228, Oakland, California, May 2000.

[19] A new probabilistic rekeying method for secure multicast groups Shankar Joshi, Alwyn R. Pais,

[20] Bandwidth Efficient Key Distribution for Secure Multicast in Dynamic Wireless Mesh Networks, Seungjae Shin, Junbeom Hur, Hanjin Lee, Hyunsoo Yoon WCNC 2009 proceedings.

[21] Joe Prathap P M. , V.Vasudevan,"Analysis of the various key management algorithms and new proposal in the secure multicast communications", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No.1, 2009

[22] Daniele Micciancio and Saurabh Panjwani, "Optimal Communication Complexity of Generic Multicast Key Distribution", IEEE/ACM Transactions on Networking (2008).

[23] Daniele Micciancio and Saurabh Panjwani, "Optimal Communication Complexity of Generic Multicast Key Distribution", IEEE/ACM Transactions on Networking (2008).

[24] Lein Harn and Changlu Lin , "Authenticated Group Key Transfer Protocol Based on Secret Sharing", IEEE transactions on computers, vol. 59, no. 6, June 2010

[25] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably-Secure Authenticated Group Diffie-Hellman Key Exchange," ACM Trans. Information and System Security, vol. 10, no. 3, pp. 255-264, Aug. 2007.

[26] J.C. Cheng and C.S. Laih, "Conference Key Agreement Protocol with Non-Interactive Fault-Tolerance Over Broadcast Network," Int'l J. Information Security, vol. 8, no. 1, pp. 37-48, 2009.

[27] K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai, and T.S. Chen, "A Conference Key Agreement Protocol with Fault-Tolerant Capability," Computer Standards and Interfaces, vol. 31, pp. 401-405, Jan. 2009