IPHV7I3022X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

KEY MANAGEMENT SCHEME FOR SECURE GROUP COMMUNICAION IN MOBILE NETWORKS

[#1]HITESH PRAKASH PATIL, Pursuing M.Tech,
[#2]K.V. BHOSALE, HOD, Associate Professor,
Dept of CSE
MIT COLLEGE, AURANGABAD (MS), INDIA.

Abstract: For any multicast group communication, group key agreement was found to be challenging because of its dynamic nature. Group key management scheme are of either distributed, centralized or hybrid architecture. Although many solutions have been proposed to handle group key changes, this paper gives the aspects of rekeying performed by those schemes. In the entire existing scheme the primary security mechanism of group communication is achieved by conventional encryption algorithms, in which key distribution and rekeying of the group key was done by Group controller. This paper explores the various algorithms along with the performances and derives an improved method. In this paper we propose a distributed key distribution scheme, using a logical group key structure, PFMH tree, and the concept of virtual user position. This approach ensures the forward secrecy and backward secrecy, reduces the rekeying complexity, communication, computation and storage complexity and time cost. Multicast communication is an efficient method of disseminating data to a group of beneficiaries over an open access network. Secure distribution of information to authorized recipients is an important prerequisite for group applications with commercial potential. An ever-increasing number of Internet applications, such as content and software distribution, distance learning, multimedia streaming, teleconferencing, and collaborative workspaces, need efficient and secure multicast communication. However, efficiency and security are competing requirements and balancing them to meet the application needs is still an open issue. A scalable group communication model ensures that whenever group membership changes the confidentiality of the group is to be maintained. This paper addresses the growth of secure group communications over a decade.

Keywords: ***Secure Group communication, Session Key, Rekeying, Multicast, tree*-based key distribution, multicast key distribution schemes.**

I.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". There has been a growing demand in the past few years for security in collaborative environments deployed for emergency services, as well as many applications including military, business, government and research organization. Examples of such collaborative applications include tele/videoconferencing, white-boards, distributed simulations, stock quote system, Pay TV, and cloud groups. All these applications depend on a framework called as secure group communication. Group communications employs IP multicasting in order to deliver the data to n receivers using a single message. Though the communication overhead is greatly reduced compared with unicasting , the security requirements becomes a big challenge. A primary method of limiting the accesses to the group data is by encrypting the contents intended for the group with a common key called as session key or group key. Since the encryption algorithm is open to all, the security relies entirely on the selected group key.

Furthermore the group key should be renewed for every change in group membership. This process is coined as Group Rekeying. The rekey policy of each group is unique but it should ensure the following security requirements.

1. Forward Secrecy
2. Backward Secrecy
3. Collision Resistance

The remainder of the article is organized as follows. The different requirements and services of a security system of a group communication is analyzed in section II. The various parameters which measures the performance of the group communication services are discussed in Section III. Section IV discusses about the diverse framework used for group communication and the various group rekeying policies. At the end the challenges and application of group communication are summarized. The forward secrecy ensures that the members who left the group cannot get access to future group data, and the backward secrecy ensures that currently joined members cannot access past group data. For a multicast group with a large number of members, key-tree based schemes were introduced to decompose a large group into multiple subgroups with smaller sizes [4], [5], [6], [7], [8]. Using these schemes, communication complexity is reduced at the cost of increase in storage and computation complexity, very few efforts

IPHV7I3022X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

have been made to reduce computation complexity, Communication complexity, storage complexity, Time cost and Scalability.

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. In any key distribution schemes, a basic operation is needed to distribute a piece of secret data to a small group of members, where each member shares a different individual key with the GC. In all current existing schemes, this operation is fulfilled by the GC using conventional encryptions followed by unicasts. A new scheme called efficient computation multicast key distribution [1] realizes this operation using one erasure decoding of certain MDS code, followed by one multicast to all he members and centralized in nature which uses MDS code- Based Rekeying on the key tree. This key tree based rekeying does not change communication complexity and storage complexity [9].

II. RELATED WORK

Several good explorations have been done for dealing with the group key distribution in a large group with frequent membership changes. There are two types of key establishment protocols: key transfer protocols and key agreement protocols. Key transfer protocols rely on a mutually trusted key generation center (KGC) to select session keys and then transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration.

In key agreement protocols, all communication entities are involved to determine session keys. The common key agreement protocol used in most distributed group key management protocols is Diffie-Hellman (DH) key agreement protocol. Some of the examples are: Bresson et al. [25] constructed a generic authenticated group DH Key exchange and the algorithm is provably secure. Katz and Yung [28] proposed the first constant-round and fully scalable group DH protocol which is provably secure in the standard model. The main feature of the group DH key exchange is to establish a secret group key among all group members without relying on a mutually trusted KGC.

Secure Group communication (SGC) expects the following as the basic security needs for any applications.

1. Group Authentication – It deals with identifying the members as legitimate members and non group members. The basic authentication policy for any group framework is to allow only the legitimate members of the group to access the current group data and able to authenticate the source for the data.

2. Group Admittance Management : It specifies the level of access and permissions for group resources to each members of the with the help of access control list.

3. Group Secrecy: The most important security requirements for any group are group secrecy which ensures the communication inside the group is only by the group members in a confidential manner. The messages intended for the group is encrypted with a secret key known as group key. The group key is known to only members of the group.

Group Survivability: It ensures that the members of the group can bale to access any group data in the presence of an attacker.

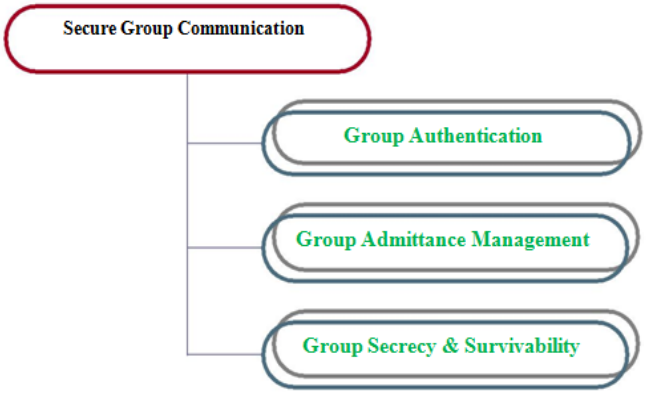

Fig 1. Requirements for Secure Group Communication

III. PERFORMANCE MEASURES OF SGC

This section illustrates the fundamental attributes of secure group communication and lists out the performance measurement indicators for any secure group communication.

1. Type of Group: Generally the group can be open group or closed group. The open group does not provide any admission control and the closed group allows only the authorized members of the group. For any type of group the group management framework is divided into three different ways

a. Centralized Group Management Architecture

b. Distributed Group Management Architecture

c. Decentralized Group Management Architecture

2. Types of Overhead : All types of group communication network incurs three different types of overhead

a. Storage Overhead: Amount of memory required to store the list of group members and the group key and other cryptographic materials by the members and the controller of the group.

b. Communication Overhead: The overhead involved in communicating the new session key and other cryptographic material in order to ensure group confidentiality.

c. Computation Overhead: It deals the process overhead involved in maintaining group secrecy.

3. Scalability: Group size should affect the performance of the group . The group should be scalable enough.

4. Resilience: It deals with the types of security threat model built to survive from any attacks. Generally the security threat model may be based on the types of attacks.

5. Most famous attacks are network based attacks and service based attacks. Attacks explores the vulnerabilities of the network.

III. COMPARISON OF GROUP KEY MANAGEMENT PROTOCOLS

3.1. 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 of the centralized entity could be used to provide fault tolerant and/or load sharing [11]. Some examples of centralized group key protocols are: Logical Key Hierarchy (LKH) [12], One-Way Function Trees (OFT) [13] and Key Management using Boolean Function Minimization (KM-BFM) technique [14]. In LKH protocols, the key server stores (2m + 1) symmetric keys and each member stores (log2m+ 1) symmetric keys. OFT proposed a variation of LKH by employing a functional relationship among the node keys in a binary key tree along the path from the leaf node representing the leaving member to the root. OFC reduces the communication overhead from LKH's 2 log2 N – 1 to log2 N by introducing a public pseudo-random function G which doubles the size of its input [21]. In KM-BFM protocol, the key server stores 2(log2 m+1) keys and each member stores (log2m + 1) keys. The centralized approaches are generally based on the idea of LKH where a key distribution center maintains a key tree. Each member knows all the symmetric keys from its leaf to the root.

3.2 Decentralized Key Management Protocols:

The management of a large group is divided among subgroup managers, trying to minimize the problem of concentrating the ork 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.

3.3 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 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.

IV. GROUP COMMUNICATION FRAMEWORK

The security of the group communication mainly resides on the group key or session key. Since the group membership is dynamic it is essential to change the group key for each member join and leave. The process of changing the group key is coined as Rekeying. The rekeying algorithm should meet the following requirements:

1. Forward Secrecy: It assures that a submissive challenger who knows a contiguous subset of old session keys cannot ascertain any subsequent session key.

2. Backward Secrecy: It ensures that a submissive challenger cannot ascertain former session key by knowing only the present session key.

3. Perfect forward Secrecy: Ensures that a compromise of a long-term key seed that generates the present short-term key(s) cannot deprive the secrecy of other previous short-term keys which have been generated by the compromised long-term key.

4. Collusion resistance: Unfeasibility for any two or more former group members who have been expelled, to gain access to future group keys even if they collude and put their keying material jointly.

Based on the rekeying policy the secure group communication framework is classified into

a. Centralized Group Key Management Architecture
b. Distributed Group Key Management Architecture
c. Decentralized Group Key Management Architecture

A. ***Centralized Group Key Management Architecture:***

In this architecture only one entity controls the entire group called as Group Controller (GC). Since it is an independent unit it does not wait for any other to generate and communicate the new key. But it suffers with a major problem of "1 affects n". If the GC fails then the total group function abruptly stops. The overall performance depends only on the group controller. And also the scalability is a major issue. The efficiency of the protocols under this are measured by size of the rekeying messages. The various benchmark protocols under this are

1. GKMP (Group Key Management Protocol)
2. LKH (Logical Key Hierarchy)
3. OFT (One-way Function Tree)
4. OFCT (One-way Function Chain Tree)
5. FT (Flat Table)
6. ELK (Efficient Large group Key)

IPHV7I3022X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

B. ***Decentralized Group Key Management Architecture***

The whole is subdivided into various subgroups and each subgroup is controlled by a central entity in the subgroup. Different controllers used are to avoid the problem of single point of failure. It supports the graceful degradation. In addition to the standard performance measures the following are also used in this category.

a. Decentralized Controller
b. No of Subgroups
c. Rekey Policy in a subgroup

The various benchmark protocols under this are

1. SMKD (Scalable Multicast Key distribution)
2. Iolus
3. DEP (dual Encryption Protocol)
4. CS (Cipher Sequences)
5. Marks
6. Kronos
7. IGKMP(Intra-domain Group Key Management)
8. Hydra

C. ***Distributed Group Key Management Architecture***:

It follows the principle of no group controller. The Group key is generated in a contributory fashion by all the members of the group or any one member of the group. It is fault tolerant. But the computation time increases linearly with the group member. It is very important to ensure the integrity of the rekey messages. In addition to the standard performance measures the following are also used in this category.

a. Processing during setup
b. Number of rounds
c. Number of Rekey Messages

D. ***Group Rekeying Algorithms***

The process of changing the group key for members join and leave is termed as group rekeying. Group Rekeying can be classified into

1. Immediate Rekeying
2. Batch Rekeying
3. Exposure Oriented Rekeying

Immediate Rekeying: Algorithms which ensures that for each change in group membership the rekeying mechanism is executed. This is suitable for application like military communication

For large and dynamic groups implementing ideal forward and backward secrecy may be infeasible because of the increased cost of rekeying. The standard approach is to amortize the rekeying cost over multiple join / leave. With this idea Batch and Exposure oriented rekeying algorithms are proposed.

Batch Rekeying Algorithm: Individual rekeying is inefficient due to overhead involved in computing and communicating the new key for every change in membership. It also induces the problem called as "Out of sync" problem. Batch rekeying algorithms can be ideal solution for application where forward and backward secrecy can be relaxed for while.

Exposure Oriented Rekeying: In this rekeying, the total number of join and leave is collected when it crosses the limit defines as threshold value the rekeying algorithm is executed.

The experiments are carried out on an Intel Core 2Duo 2.80- GHz machine with a 2-Gbyte memory running windows XP. The implementation results of computations and communications are presented in Fig.3 and 4. From these results; we can see that upon a single-user join event, PFMH has the lowest cost among all the schemes. Compared with GC, PFMH has more than 10 percent reduction in computation cost and a more than 65 percent reduction in communication cost and time cost. Compared with GC, the reduction is even more, about 50 percent in computation cost and about 80 percent in time and communication costs. Upon a single-user leave event, compared with GC, PFMH has about a 25 percent reduction in computation cost, about a 15 percent reduction in time cost, and a similar communication cost. Although PFMH has slightly higher computation and communication costs than GC upon a single user leave event, when averaged over both join and leave events, the reduction is still significant, with a 20 percent reduction in computation cost, 35 percent reduction in communication cost, and 40 percent reduction in time cost. Fig 5 and 6, shows the key distribution time and key recovery time of both the scheme under various multicast group sizes. It is clear that using one- way hash functions adds none-trivial computation complexity. Nevertheless, the proposed scheme still out performs the GC schemes by a significant margin. The computation time of the key distribution is also compared to GC for a selected multicast group size. Notice that the computation times of both the GC is significantly larger than proposed schemes.

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

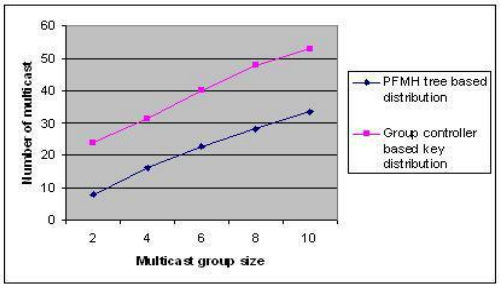

Fig. 3 Computation cost

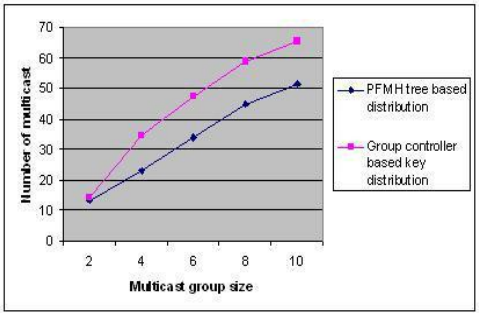

Fig. 4 Communication cost

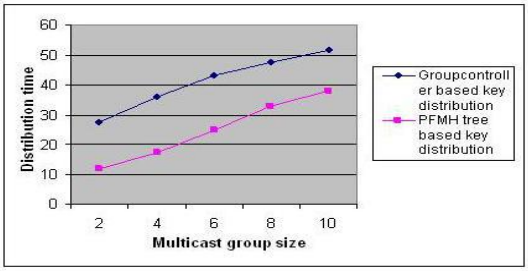

Fig.5 key distribution time

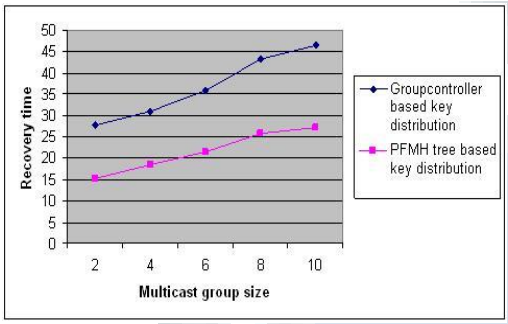

Fig.6 key recovery time

V.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. The MDS codes was combined with PFMH trees and performance of distribution time and key recovery time was evaluated, this scheme provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for dynamic group key distribution. This scheme is thus practical for many applications in various broadcast capable networks such as Internet and wireless networks.

In this paper a detailed survey is presented in the area of secure group communication. First the security requirements are discussed. The difference performance measurements indicators are explained. The classification of group key management algorithms and the benchmark algorithms are compared for each category. The different rekeying policies and the application where these rekeying algorithms can be employed are also discussed.

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

[24] 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.

[25] 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.

[28] J. Katz and M. Yung, “Scalable Protocols for Authenticated Group Key Exchange,” J. Cryptology, vol. 20, pp. 85-113, 2007.

[29] Wei Yu, Yan (Lindsay) Sun, Member, IEEE, and K.J. Ray Liu, Fellow, IEEE, “Optimizing the Rekeying Cost for Contributory Group Key Agreement Schemes”, IEEE transactions on dependable and secure computing, vol. 4, no. 3, July-September 2007.