International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3 Issue 7, July 2014

Securing the Multicast Groups in Wireless Networks

Miss. S.Vijayashanthi[1], Mr. S.Balamurugan[2]
[1] PG Student, Sri Manakula Vinayagar Engineering College, Pondicherry-605106
[2] Assistant Professor, Sri ManakulaVinayagar Engineering College, Pondicherry-605106

***Abstract*— In computer networking, multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source. However, the existing group key management (GKM) schemes, aiming to secure communication within a single group, are not suitable in multiple multicast group environments because of inefficient use of keys, and much larger rekeying overheads. In this paper, we propose a new GKM scheme for multiple multicast groups, called the master-key-encryption-based multiple group key management (MKE-MGKM) scheme. The MKE-MGKM scheme exploits asymmetric keys, which are generated from the proposed master key encryption (MKE) algorithm, and is used for efficient distribution of the group key. It alleviates the rekeying overhead by using the asymmetry of the master and slave keys, i.e., even if one of the slave keys is updated, the remaining ones can still be unchanged by modifying only the master key. Through numerical analysis and simulations, it is shown that the MKE-MGKM scheme can reduce the storage overhead of a key distribution center (KDC) by 75% and the storage overhead of a user by up to 85%, and 60% of the communication overhead at most, compared to the existing schemes.**

***Index Terms*— group key management, multicast, and master key encryption.**

I. INTRODUCTION

The multicast group can be identified with the class D IP address so that the members can enter or leave the group with the management of Internet group management protocol. The trusted model gives a cope between the entities in a multicast security system. For secure group communication in the multicast network, a group key shared by all group members is required. This group key should be updated when there are membership changes in the group, such as when a new member joins or a current member leaves. Along with these considerations, we take the help relatively prime numbers and their enhancements that play a vital role in the construction of keys that enhances the strength for the security.

Manuscript received July, 2014.
S.Vijayashanthi, *MCA Department, Sri Manakula Vinayagar Engineering College, Pondicherry, India, 8754931818,*
S.Balamurugan, *MCA Department, Sri Manakula Vinayagar Engineering, Pondicherry, India, 9865605523,*

Multicast cryptosystems are preferably for sending the messages to a specific group of members in the multicast group. Unicast is for one recipient to transfer the message and "Broadcast" is to send the message to all the members in the network. Multicast applications have a vital role in enlarging and inflating of the internet. The Internet has experienced explosive growth in last two decades. The number of the Internet users, hosts and networks triples approximately every two years. Also Internet traffic is doubling every three months partly because of the increased users, but also because of the introduction of new multicast applications in the real world such as video conferencing, games, ATM applications etc... Broad casting such as www, multimedia conference and E-commerce, VOD (Video on Demand), Internet broadcasting and video conferencing require a flexible multicasting capability. Multicast is a relatively new form of communications where a single packet is transmitted to more than one receivers. The Internet does not manage the multicast group membership tightly. A multicast message is sent from a source to a group of destination hosts. A source sends a packet to a multicast group specifying as the multicast group address. The packet is automatically duplicated at intermediate routers and any hosts that joined the group can receive a copy of the packet.

II. LITERATURE SURVEY

S. Anahita Mortazavi, And Alireza Nemaney Pour [1] This paper presents an efficient group key management protocol, CKCS (Code for Key Calculation in Simultaneous join/leave) for simultaneous join/leave in secure multicast. This protocol is based on logical key hierarchy. In this protocol, when new members join the group simultaneously, server sends only the group key for those new members. Then, current members and new members calculate the necessary keys by node codes and one-way hash function. A node code is a random number which is assigned to each key to help users calculate the necessary keys. Again, at leave, the server just sends the new group key to remaining members.

Firdaus Mah [2] This paper will examine the main components of the trust model and GSA. Base on that, the paper will layout the architecture and design requirements needed for group key management protocol. There are several group key management protocols that are proposed, the paper will however elaborate mainly on Group Security Association Key Management Protocol (GSAKMP).

An important component for protecting group secrecy is rekeying. In the event of the group being

ISSN: 2278 – 1323

compromise due to members leaving and joining the group or an unauthorized access to a cryptographic key, a secure group key management protocol requires to handle it by performing rekeying operation. The paper will explain on a rekey algorithm called Logical Key Hierarchy (LKH) that is generally recommended for performing group rekeying operation.

Chung Kei Wong, Mohamed Gouda [3] Many emerging network applications (e.g., teleconference, information services, distributed interactive simulation, and collaborative work) are based upon a group communications model. As a result, securing group communications, i.e., providing confidentiality, authenticity, and integrity of messages delivered between group members, will become a critical networking issue. We present, in this paper, a novel solution to the scalability problem of group/multicast key management. We formalize the notion of a secure group as a triple where denotes a set of users, a set of keys held by the users, and a user-key relation. We then introduce key graphs to specify secure groups. For a special class of key graphs, we present three strategies for securely distributing rekey messages after a join/leave and specify protocols for joining and leaving a secure group.

David A. McGrew and Alan T. Sherman [4] We present and analyze a new algorithm for establishing shared cryptographic keys in large, dynamically changing groups. Our algorithm is based on a novel application of one-way function trees. In comparison with previously published methods, our algorithm achieves a new minimum in the number of bits that need to be broadcast to members in order to re-key after a member is added or evicted. The number of keys stored by group members, the number of keys broadcast to the group when new members are added or evicted, and the computational efforts of group members, are logarithmic in the number of group members. Our algorithm provides complete forward and backwards security: newly admitted group members cannot read previous messages, and evicted members cannot read future messages, even with collusion by arbitrarily many evicted members.

III. METHODOLOGY

There are five methodologies used for the group communication that are used to perform the encryption and decryption process.

A. User Interface Design

In this module, Design the interface for the user to interact with the application. Each and every software application needs fine steps interact with the end users of the application. The information's are stored in database. When the user registration, same user cannot register more than one time. The unique user only allowed for key allocation. The users login with the specified id and password to access their schema information.

B. Multicast Service Module

In order to implement the multicast, i.e., the delivery of data only to the members of a group, in wireless networks, we need to an access control mechanism for the broadcasted messages, which guarantees confidentiality, protects digital contents, and facilitates accurate accounting. In multicast module checks the user authentication and then it transmits the messages to the user.

C. Key Allocations

In this module, we obtain key size, using key allocation algorithms. That is how many public keys and private keys allocated based on network size (number of user). After allocation keys, generate the distinct private key sets those who are all registered in Authentication server. Each user stored the common public keys and an own private key set. The key can be allocated based on the number of user in the group.

D. Encryption

In this module, encrypt the messages using the public key. Every user in mission critical environment is able to communicate securely with other user, with the help of their stored keys. Before Encryption, the user to make a request ID to the user whom is going to send message.

After that, the public keys would be getting for Encryption based on receiver ID (Binary value). Here, the sending message would be encrypted using public key one, and then cyber text is encrypted one more time using public key two. Finally the message is transmitted to destination user. The usual way to provide an access control mechanism for the secure group communication is to employ a symmetric key, known as a public key, shared only by group members.

E. Decryption

In this module, Decrypt message using already stored private keys set. First Decrypt the message using private key one and then to make another decryption using second private key. Finally we can show message in receiver Text Area. The private key can be used to decrypt messages, which can be encrypted by only one public key or to decrypt messages encrypted with several different public keys. The individual key can be provided for the entire user. The usual way to provide an access control mechanism for the secure group communication is to employ an asymmetric key, known as a private key.

IV. EXISTING SYSTEM APPROACH

The existing GKM schemes still face the limitation of rekeying performance as the number of multicast services increases. However, in the foreseeable future, multiple multicast groups will co-exist in a single network due to the emergence of many group-based applications. In such a situation, it is likely that the service provider may suffer from considerable key management overhead for supporting multiple multicast groups.

In existing system if the user wants to access from different server the user have to login each time for different server hence the key management process become too tedious to handle if the server number increases tremendously Once the master key has been change the entire configuration has to be configured to access the service.

ISSN: 2278 – 1323

V. PROPOSED APPROACH FOR MGKM

Prior to the description of the proposed scheme, the MKE scheme is presented. An asymmetric KEK, that can alleviate the rekeying cost, is made by the MKE scheme.

Alternative for MGKM: MKE Scheme

In the MGKM, a user can belong to one or more groups. Since the membership changes of a user having multiple group keys can affect all users of the corresponding groups, the effect of 1-affects-n may become much more severe.

The reason why the existing schemes cannot efficiently alleviate the rekeying cost is that they use only symmetric key encryption for both the TEK and the KEK. In contrast, a new asymmetric key scheme, called the MKE scheme, has been introduced in KEKs to solve this problem. Since the TEK is used to encrypt/decrypt every packet, it should be based on symmetric key encryption that is much faster than asymmetric key encryption. However, since the KEK is used only for distributing the TEKs, it is reasonable that an asymmetric key is used as a KEK.

The MKE scheme is a RSA-based public-key cryptosystem proposed by Koyama where every user in the RSA system has a key pair that consists of a public key and a private key, each of which is used for encryption and decryption in an asymmetric pair wise manner. The master key can be used to encrypt messages, which can be decrypted by several different private keys or to decrypt messages encrypted with several different public keys. The most important feature of the MKE for MGKM is that one of the key pairs can be easily changed by modifying only the master key, without any changes to other users' key pairs.

VI. DESIGN

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

The Figure 6.1 shows the architectural design of the overall system. It represents how the message can be encrypted and decrypted by the user and admin.

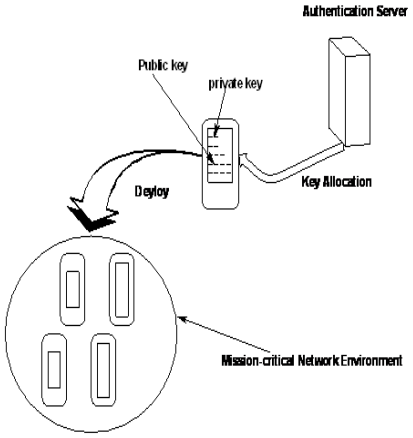


Fig.:6.1 Architecture Diagram

VII. EXPERIMENTS RESULTS AND PERFORMANCE COMPARISONS

In this section, we compare CKCS protocol with some previously proposed ones, LKH, OFT, and OKD. We compare these protocols at join and leave operations for simultaneous mode. The comparison measures are based on key generation, key encryption, communication overhead, and message size.

Tables 1, 2, 3 and 4 summarize our comparisons, focusing on the following measures:

1) Computational overhead

a) Key generation overhead: The number of keys that must be generated at join/leave.

b) Encryption overhead: The number of encryptions.

2) Communication overhead: The number of transmissions from the key server.

3) Message size: the total number of keys in one message.

The computational overhead is the sum of key generation and key encryption. In our comparisons shown in the following tables, *n* denotes the group size which is the number of members in the multicast group after join and before leave operations.

In addition, in simultaneous mode, *m* denotes the number of members that join or leave multicast group concurrently. Finally, in simultaneous mode we consider that *m* _ *n*. In other words, the number of simultaneous users is less than or equal to the number of group members. Binary key tree is assumed for key degree in comparison. As stated before, in binary key tree the height of tree is *log2n* which shows the number of nodes in each branch. Obviously, the efficiency of a protocol is related to the height of the key tree. In other words, a key tree with smaller height is more efficient than a tree with larger height.

ISSN: 2278 – 1323

Consequently, since the tree height for all of these protocols is equal, the factors that make differences in decreasing overhead are the re-keying procedure and the key distribution technique. The re-keying method itself is an effective way to reduce the overhead. Regarding re-keying method in LKH, OFT, and OKD, the server has more loads for generating, encrypting, and delivering keys to the members. In LKH and OFT, the members do not participate in key calculation on each membership changes. While in OKD the members involve key update process with the key server but the re-keying overhead problem for new members still remains.

Conversely in CKCS, the contribution of current members in re-keying process and minimum number of key delivery to new members are two decisive factors which make it more effective than the other ones. Finally, the previously proposed protocols do not consider the simultaneous mode at all. So, the overhead of these approaches is not acceptable for this mode. Assuming simultaneous mode for these protocols, the results are shown in Tables 1, 2, 3 and 4.

In addition, we have implemented programs to compare the overhead of these mentioned protocols. These programs compare the processing time that each protocol spends for generating and encrypting necessary keys after each membership changes based on the number of users. Finally, we sketch some plots (Figures 11, 12, 13, 14, 15, and 16) for showing numerical comparison. For this purpose, we consider that there are 100,000 group members and 1024, 2048, 4096, and 8192 simultaneous users for join/leave.

VIII. COMPUTATIONAL OVERHEAD

The computational overhead for these Protocols depend on the number of keys that need to be generated and encrypted by the server.

Table 1 shows the key generation overhead at simultaneous join/leave operation. In LKH, group members do not participate in middle node keys calculation in each join/leave operation. In OFT, only a new member at join and the remaining members at leave need to update the keys in their paths. In OKD, when a member joins the group all the necessary keys should be delivered to him/her by unicast and all the remaining members can update their middle node keys by themselves, but at leave some nodes are responsible for updating the affected keys. So, in these protocols, the key generation overhead is *log2n* for a single member and *mlogn* when *m* members join/leave the group simultaneously. CKCS has the smallest overhead for key generation comparing with the others.

Table 1.The comparisons of key generation overhead in simultaneous join/leave operations.

Key Generation		
Protocols	**Join**	**Leave**
LKH	mlog^2n	mlog^2n
OFT	mlog^2n	mlog^2n
OKD	mlog^2n	mlog^2n
CKCS	m+1	1

As mentioned above, the previously proposed protocols do not consider the simultaneous mode. According to the results, in LKH, OFT, and OKD when *m* members join/leave the multicast group, the server generates keys to update the key tree (all the keys in the paths of *m* members to the root). If *m=1*, these amounts are equal to the single mode.

Table 2 illustrates the key encryption overhead for *m* simultaneous join/leave. The results show that LKH, OFT and OKD have high overhead at join/leave. But in CKCS, this overhead is the lowest one because in each join operation the server encrypts only the new group key with each new member's individual key. In CKCS, the key encryption overhead at leave is equal to the height of the key tree because the server encrypts the group key by the top node of each part for users who are located at that part.

Table 2.The comparison of key encryption overhead at simultaneous join/leave operations.

Figures 7.1 and 7.2 illustrate the computational overhead (processing time versus the number of simultaneous users) for simultaneous join/leave. As shown, LKH has the highest gradient when *m* members join/leave the group concurrently. OFT and OKD have the lower computational overhead than LKH in simultaneous join/leave. In simultaneous join operation, the computational overhead of OFT is higher than OKD but at leave these protocols have almost the same overhead. CKCSS has the lowest overhead at both simultaneous membership changes.

Key Encryption		
Protocols	**Join**	**Leave**
LKH	3mlog^2n	2mlog^2n
OFT	2mlog^2n	mlog^2n
OKD	mlog^2n	mlog^2n
CKCS	m	Log^2n

ISSN: 2278 – 1323

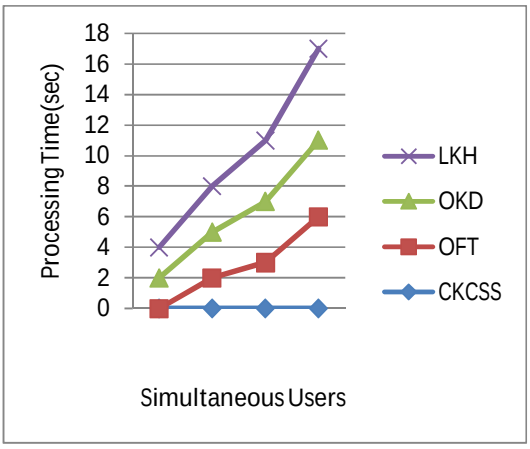


Figure 8.1 Computational overhead versus number of group members at simultaneous join.

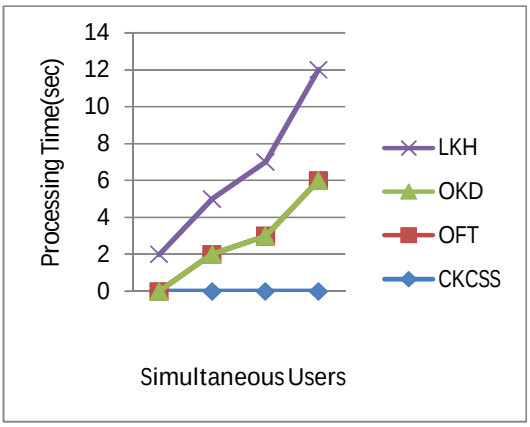


Figure 8.2 Computational overhead versus number of group members at simultaneous leave.

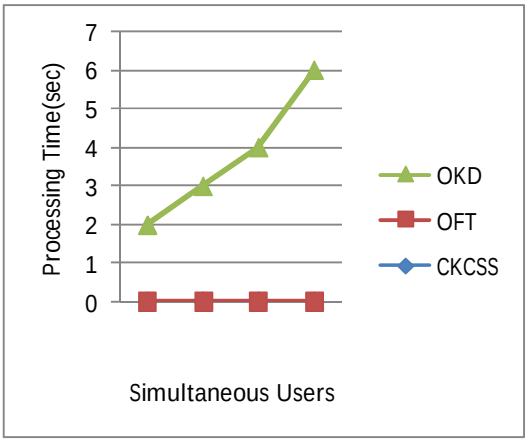
IX. COMMUNICATION OVERHEAD AND MESSAGE SIZE

Table 3 depicts the communication overhead at simultaneous join/leave operation. Communication overhead at simultaneous join is divided into two categories, unicast and multicast overhead.

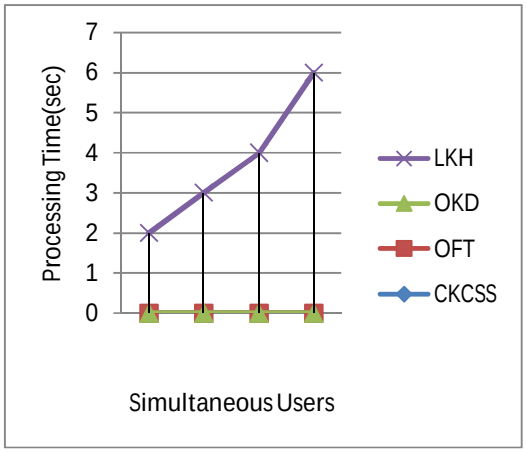
As shown in this table, LKH and OFT have the same communication overhead at join/leave which is the highest one. These two protocols have both unicast and multicast communication in each simultaneous join. This happens because necessary keys for new members are sent by unicast and for remaining members by multicast. In OKD and CKCS, all the keys are collected in one message, and sent to new members by one multicast message. So, there is no overhead for unicast but *1* multicast transmission exists when *m* members join the group. Figures 7.3.1 and 7.3.2 illustrate the numerical results for communication overhead at simultaneous join/leave respectively.

Table 3.The communication overhead at simultaneous join/leave operations.

Communication Overhead			
Protocols	**Join**		**Leave**
	Unicast	**Multicast**	**Multicast**
LKH	nlog^2	n2log^2n	nlog^2n
OFT	nlog^2n	nlog^2n	nlog^2n
OKD	nlog^2n	-	nlog^2n
CKCS	-	1	nlog^2n



(a)



(b)

Figure 9.1 Communication overhead versus number of group members simultaneous join

(a) Unicast communication (b) multicast communication.

ISSN: 2278 – 1323

Table 4 shows message size in simultaneous join/leave. For simultaneous join/leave, CKCS has the lowest message size in each message transmission. All the other protocols have large message size for simultaneous because of their necessary transmissions. In CKCS, the server sends one multicast message which includes *m* keys. Each of these keys contains the group key encrypted by the individual key of each simultaneous user.Figures15 and 16 illustrate the numerical results for message size at simultaneous join/leave respectively.

Table 4.Message size at simultaneous join/leave operations.

Message Size		
Protocols	**Join**	**Leave**
LKH	2mlog^2n	2mlog^2n
OFT	mlog^2n+1	mlog^2n+1
OKD	mlog^2n	mlog^2n
CKCS	m	m

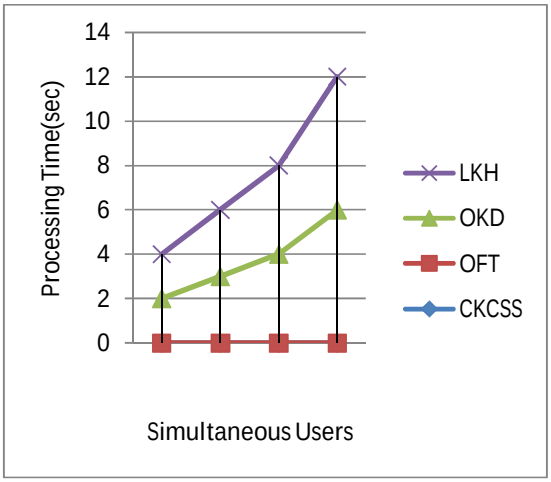


Figure 9.2 Message size at simultaneous join

Although LKH based protocols minimized the overhead of leave operation to *log2n*, they added unnecessary overhead to join operation. This amount gets larger when number of users increases. With a glance at Tables 1, 2, 3, and 4 it is not difficult to see that CKCSS has two major features. First, the overhead of CKCSS at join does not depend on the number of users. It means that the overhead for new member is a constant amount while there is no overhead for current users. Second, reducing the overhead for new user at single join is the other important factor for simultaneous mode.

X. CONCLUSION

In this paper, an MGKM scheme has been proposed that can enhance the management performance of multiple group keys regardless of the hierarchy of the users or the data streams. In contrast to other existing schemes using only symmetric keys, the MKE-MGKM scheme exploits asymmetric keys, i.e., a master key and multiple slave keys, which are generated from the proposed master key generation algorithm. By using a set comprising a master key and slave keys, a TEK can be efficiently distributed to multiple SGs. Therefore, the number of rekeying messages can be dramatically reduced. Also, since the key graph of the MKE-MGKM scheme is much simpler than that of other schemes, less memory is needed for storing the keys. Compared with other schemes, the MKE-MGKM scheme can significantly reduce the storage and COs in the rekeying process, with acceptable computational overhead. It is expected that the MKE-MGKM scheme can be a practical solution for various group applications, especially for those requiring many SGs, such as TV streaming services charged on a channel by channel basis.

ACKNOWLEDGMENT

I am extremely grateful to my project guide **Mr.S.Balamurugan,** Assistant Professor, Department of MCA, Sri Manakula Vinayagar Engineering College, Pondicherry for his ubiquitous cooperation in this project.

I am extremely grateful to coordinator **Mr.A.Martin,** Associate Professor and HOD Department of MCA, Sri Manakula Vinayagar Engineering College, Pondicherry who showed keen interest in my project. This has really motivated me to work better.

I would like to express my heartfelt thanks to our project coordinator **Mr.R.Ramakrishnan,** Associate Professor, Department of MCA, his guidance from the starting of the project was really grateful without which we might not have been able to succeed in our work.

I wish to thank all the Faculty Members of the Department of MCA, Sri Manakula Vinayagar Engineering College for providing their support and guidance for my Project work.

I would like to express my faithful thanks to my beloved Director, **Dr.V.S.K.Venkatachalapathy, Sri Manakula** Vinayagar Engineering College, for having extended all the facilities of the Department without any hesitation.

I also express my gratitude to my Parents for their support, motivation and encouragement in effectively getting through the Project and the course.

Above all, the inevitable presence of God throughout is a non negotiable fact. The source of wisdom and might, the Almighty is to whom I bow my head for such a momentous dream come true.

REFERENCES

[1] IEEE Standard 802.16-2004, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE, 2004.

ISSN: 2278 – 1323

[2] Third Generation Partnership Project, "Multimedia Broadcast/ Multicast Service; Stage 1 (Release 8)," Technical Specification 3GPP TS 22.146 v.8.3.0 (2007-06), June 2007.

[4] D.M. Wallner, E.J. Harder, and R.C. Agee, "Key Management for Multicast: Issues and Architectures," IETF RFC 2627, http://www.ietf.org/rfc/rfc2627.txt, June 1999.

[5] Y. Challal and H. Seba, "Group Key Management Protocols: A Novel Taxonomy," Int'l J. Information Technology, vol. 2, no. 1, pp. 105-118, 2005.

[6] S. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.

[7] Y. Sun and K.J.R. Liu, "Hierarchical Group Access Control for Secure Multicast Communications," IEEE/ACM Trans. Networking, vol. 15, no. 6, pp. 1514-1526, Dec. 2007.

[8] C.K. Wong, M.G. Gouda, and S.S. Lam, "Secure Group Communications Using key Graphs," ACM SIGCOMM Computer Comm. Rev., vol. 28, pp. 68-79, 1998.

ISSN: 2278 – 1323