Current Trends in Group Key Management

R. Siva Ranjani[1]
Research Scholar, Dept.of CS&SE
Andhra University, Visakhapatnam,
Andhra Pradesh, India,

Dr.D.Lalitha Bhaskari[2]
Associate Professor, Dept.of CS&SE
Andhra University, Visakhapatnam,
Andhra Pradesh, India,

Dr.P.S.Avadhani[3]
Professor, Dept.of CS&SE
Andhra University, Visakhapatnam,
Andhra Pradesh, India,

***Abstract*—Various network applications require sending data onto one or many members, maintaining security in the large groups is one of the major obstacles for controlling access. Unfortunately, IP multicast is not providing any security over the group communication. Group key management is a fundamental mechanism for secured multicast. This paper presents relevant group key management protocols. Then, we compared them against some pertinent performance criteria. Finally, we discuss the new research directions in group key management.**

Keywords-Multicast; group key management; security member driven; time driven.

I. Introduction

With rapid growth in the internet, people using the group communication in applications such as paying TV, transmission of video and audio, updating software, military applications, video games etc. In recent decades, the focus is mainly on the security issues involved in the group communication. When the group uses the unicast communication, one sender is sending the data stream onto one group member. In multicasting, the group member is sending the data onto other group members. Security is main focused area in group communication. Group key management is the fundamental mechanism provides the security in group communication. In this, security is achieved by sharing a common key among the group members. The message packets, those are going to transmit should be encrypted with the shared key.

Group key management is mainly focusing on the key generation and distribution of key among the group members. All the group members should participate in the secure distribution, creation and revocation of the keys [1]. The communication session in group key management is managed by two entities: Group Controller (GC), responsible for key generation, distribution and rekeying for membership change and Key Server (KS), responsible for maintaining the keys and distributing the keys.

The scenario of group communication is shown in the figure 1. Each member in the group is having two keys (TEK and KEK). The TEK (Traffic Encryption Key) is used for encrypt, decrypt and authenticate the data transfer. The TEK for the group member is generated by the local manager. The KEK (Key Encryption Key) is used for encrypt the TEK. To multicast the message (m) secretly the sender encrypts the message with TEK using the symmetric key algorithm. At the receiving side, the receiver decrypts the message (m) with TEK. In the group communication, the members in the group are not fixed, members can join / members can leave the group. So we need to secure the sending message to be received by the group members at that instance. When member is leaving, the KS must generate a new TEK and distribute the key secretly to all other members except leaving one. This process is known as rekeying. From the figure, we observe that Key Server is sharing a secret key called Key Encryption Key (KEKi) with each group member. When the member is leaving, the KS generates a new TEK : TEK1, encrypted with their KEKi and sends it to all other group members except leaving one. So the leaving member does not know the new TEK1, to decrypt the future messages shared in the group.

When a new member is joined in the group, first it must be authenticated by the GC. After that, the KS checks the rights of the new member and adds the member in the future message transformation session. The KS generates a new secret KEKj and shared with the new member mj. In order to restrict the new member form past data access the KS generate the new TEK : TEK1, encrypted with KEKi and then sends to all the group members along with the new joined one.

II. Group Key Management Protocol

As defined by Menezes et al. in [2], Group Key Management is the set of techniques and procedures used for the establishment and maintenance of keys among members to form the group. According to Hutchison [3], group key management can be classified into three categories.

Centralized Group Key Management Protocols—Key distribution is achieved by a single entity i.e Key Distribution Center (KDC), also known as Central Authority (CA). The Central Authority maintains the entire group, allocates the individual KEK to group members. It is also responsible for sharing the common TEK among all the group members.

Decentralized Group Key Management Protocols — In Decentralized Group, the group is splitting into several subgroups. Each subgroup is managed by subgroup controller. In this approach, the hierarchy of sub group controllers shares the labor in transferring TEK to group members. This management will reduce the load on the KDC.

Distributed Group Key Management Protocols — In Distributed Group key management either all the group members or only one member is involved in group key generation. No group controller is present; this will improves the reliability of the overall system.

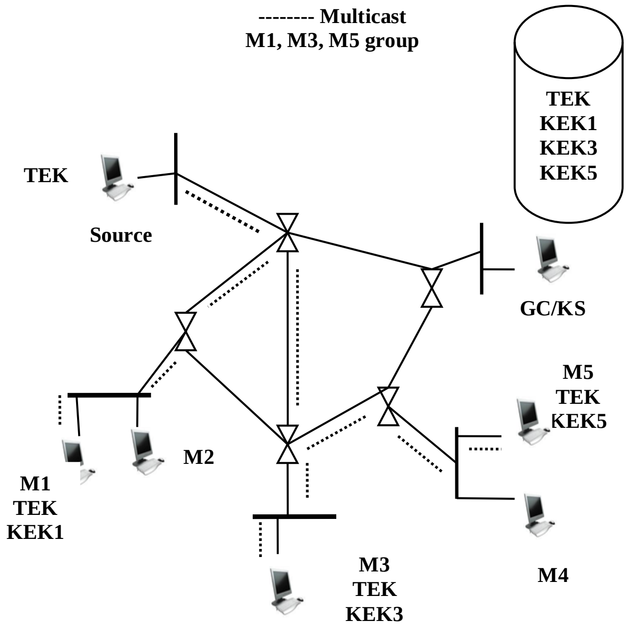

Figure 1: Group with KS and 5 Group Members

A. Centralized GKMP

Assume there is a group with n members. The Key server (KS) is the centralized group manager which stores information about all the group members. The KS takes n Key Encryption Keys (KEK) and each of them is shared with one member. The KEK is the secret key used for encrypting the group key (TEK). In the figure, the centralized group key protocols again sub divided into three categories depending on the technique used in distributing the TEK among group members.

a) Pair wise Keys:

In this sub type the KS shares a secret key called Key Encryption Key (KEK) with all the group members. This key is used for establishing the secured channel between the KS and the member for transferring the TEK securely whenever the key is required.

Harney and Muckenhirn [4],[5] proposed Group Key Management Protocol (GKMP), in this KS shares a secret key (KEK) individually with each active member and KS generates a Group Key Packet (GKP) that contains a Group TEK (GTEK) and a group KEK (GKEK). Chu et. al Protocol is proposed by Chu et. al [6], a Group leader shares a secret Key Encryption Key (KEK) with each group member.

b) Broadcast Secrets:

In this protocol, the KS broadcasts the rekey information to all the group members. Chiou and Chen [9] proposed Secure Lock protocol, in this key server uses a single broadcast to establish the group key or for sending rekey to the entire group in the case of join or leave membership.

c) Keys Hierarchy:

In pair wise key approach, the KS establishes individual secure channel with the members and uses this channel for sending the TEK updates. This mechanism increases the update message overhead. In order to reduce message overhead, the Key Server in this approach, shares a secret key with subgroup in addition to individual channel. When the member leaves the group the KS uses the sub group secret key to distribute the new TEK, which are not known by the leaved member. Nemaney et al.[26] proposed hierarchical group key management that increases the efficiency of the key. Following section describes some of the protocols using this re-key mechanism.

Wong et.al and Wallner et. al[11][14] proposed Logical Key Hierarchy (LKH) protocol. In this protocol, KS is the root of the tree and maintains a tree of keys. In this protocol, each node stores at most 1+log2(n) rekey messages.

McGraw and Sherman proposed One-Way Function Tree (OFT) protocol and this is an improvement over the LKH. Here, node's KEK is calculated by the member rather than attributed by the KS. Each node in this protocol is maintaining its blended sibling keys and its leaf secret key also maintains the blinded secret KEKs of its ancestors.

Canetti et. al proposed One-Way Function Chain Tree (OFCT) protocol [13]. This protocol works similar to OFT but, a pseudo random generator is used to generate the new KEKs rather than a one-way function, and it is done only during user removal.

Efficient Large Key distribution (ELK) [15] approach uses the pseudo random function for generating the new KEK when a membership change takes place. Waldvogel et.al [16] proposed Centralized Flat Table Key Management (CFKM) protocol, this approach, uses the flat table concept in order to reduce the number of keys maintained by the KS. Flat table consists of one TEK and 2w entries for KEKs, where w is the number of bits in identifier of a member. Wong et.al protocol is the extension of the LKH protocol [14]. The LKH uses the binary tree for key distribution; wong et.al uses the k-ary tree.

Comparison of centralized group key management protocols:

Table I compares the centralized group key management protocols. The efficiency of the protocol can be compared against the following criteria: 1 affect n, forward and backward secrecy, storage requirements at KS and group member, collusion, join re-key overhead and leave rekey overhead.

a) Decentralized Group key Management Protocols

The group members are arranged into some subgroups, and each subgroup has a controller called key manager. The key managers of the subgroup share the labor of distributing the TEK to group members in order to avoid bottle necks and single point of failure. Decentralized group key management is categorized into member ship driven and time driven re-keying.

Ballardie's Scalable Multicast Key Distribution (SMKD) protocol [18] propose a group key distribution method based on the Core Based Tree (CBT) multicast routing protocol. In CBT architecture, the multicast is rooted at main core. In Intra Domain Group Key Management (IGKMP) [17], the network divides into administratively scoped areas. This protocol is having Domain Key Distributor (DKD) and Area Key Distributor (AKD).

TABLE I : COMPARISON OF CENTRALIZED GROUP KEY MANAGEMENT

Protocol Type	Server Storage	Rekeying overhead	
		Member Join	Member leave
Poovendran et al	n+2	2	2n
Dunigan & chao	n+2	2	2n
Chu et. Al	n+2	2	2n
Secure Lock	2n	2	0
LKH	2n-1	$\log_2(n)+1$	$2\log_2(n)$
OFT	2n-1	$\log_2(n)+1$	$\log_2(n)+1$
OFCT	2n-1	$\log_2(n)+1$	$\log_2(n)+1$
ELK	2n-1	$\log_2(n)+1$	$\log_2(n)+1$
CFKM	2I+1	2I	2I

Where n: number of group members I: number of bits in a number id

The DKD is responsible for generating group TEK and is propagated to the group members through AKD. The DKD and AKDs belong to multicast group called All-KD-Group.

In Hydra protocol [19] the group is organized into sub groups. Each sub group has a server called Hydra Server (His) responsible for controlling the sub group. BAAL protocol has three entities: First is the Group controller (GC), responsible for maintaining the participant List (PL) and creating and sending the group key TEK to member through local controller. Second is Local Controller (LC), responsible for managing the keys in subnet, receives the new TEK and distributes to members connected to subnet. Third is Group member. IOLUS protocol is the frame work of a hierarchy of multicast subgroups to constitute virtual group [20]. Each subgroup is managed by a Group Security Agent (GSA), responsible for managing key inside the sub group.

Cipher Sequences is a proposed framework for multicast security [21], based on Reversible cipher sequence. The multicast tree is rooted at source and the leaves are group members. Challel et.al proposed Scalable Adaptive Key Management Scheme (SAKM) protocol. This protocol tackles the scalability issue. SAKM tackles the scalability by organizing the group into clusters.

b) Time Driven Approach

In time driven approach, the TEK is changed after specified amount of time. When the member leaves or joins in the group they will not excluded or appointed immediately, need to wait for the beginning of the new interval of time.

Briscoe proposed MARKS protocol, suggests a slicing the time length into small portions of time and uses a different key for encrypting each slice. The encryption keys occupied at leaves in BST that is generated from a single seed.

Setia et al [22] describe a scalable approach based on time-driven called Kronos. In this protocol, Setia denotes the group with a birth and death process model and discussed the model in two occasions: correlation subscriber behavior and independent subscriber. The operation of Kronos is similar to that of IGKMP. In Dual Encryption Protocol (DEP), the group is divided hierarchically into sub groups and the sub group is managed by sub-group manager (SGM). In YANG et. al Protocol [23] approach the multicast group is organized into a set of sub groups, KS manages each subgroup. The KS is responsible for rekeying the members in the subgroup periodically. Scalable Infrastructure For Multicast Key Management (SIM-KM) uses the proxy encryptions. SIM-KM uses the proxy function that converts the cipher text for one key into the cipher text for another key.

c) Comparison of Decentralized Group Key Management Protocols

In this different Group controllers are used to manage the subgroups. Table II compares the decentralized group key management protocols. Attributes that are used for evaluating the performance of decentralized protocols are key independence, decentralized controller, local rekey, key-data transformation, rekey per membership and type of communication.

TABLE II: COMPARISON OF DECENTRALIZED GROUP KEY MANAGEMENT

Protocol	Key Independent	Decentralized Controller		Local re-key	Key Vs Data	Re-key	Communication Type
		Management	Key Server				
SMKD	Yes	Yes	Yes	No	Yes	No	Both
IGKMP	Yes	Yes	Yes	No	Yes	Yes	Both
Hydra	Yes	Yes	Yes	No	Yes	Yes	Both
Baal	Yes	Yes	Yes	No	Yes	No	Both
MARKS	No	Yes	-	No	Yes	No	Both
Kronos	No	Yes	Yes	No	Yes	No	Both
DEP	Yes	No	No	No	Yes	No	Both
Iolus	Yes	Yes	Yes	Yes	Yes	Yes	1 to n
KHP	Yes	Yes	No	Yes	No	Yes	Both
Cipher Sequences	Yes	No	No	No	Yes	Yes	1 to n
SAKM	Yes	Yes	Yes	Yes	Yes	No	1 to n
Yang et al	Yes	Yes	Yes	Yes	Yes	Yes	1 to n
SIM-KM	No	No	No	Yes	Yes	Yes	1 to n

Both: 1 to n and n to n

d) Distributed Key agreement Protocols

In distributed key agreement protocol, the group members are participated in the establishment of a group key, further classified into three categories: Ring-based, hierarch based and broadcast based.

e) Ring Based

In this, cooperation of group members forms a virtual ring. In Ingemarson Et Al. protocol, all the group members are organized into a virtual ring and the Group Diffie-Hellman (GDH) protocol uses the extension of Diffie Hellman algorithm for group key generation.

f) Hierarchy based cooperation

The group members are arranged in a tree hierarchy for group key generation. In OCTOPUS, the entire group is divided into four sub groups. The leader member in the subgroup is responsible for collecting the intermediate subgroup values and calculates the intermediary DH value. Steer et. al proposed Skinny Tree (STR) protocol, uses the tree structure. The leaves associated in the tree are group members; each leaf is identified by its position. Diffie-Hellman Logical

Key Hierarchy (DH-LKH), proposed by Perrig et al.[24] is variant of STR and uses binary tree. The binary tree built from bottom to top. Distributed Logical Key Hierarchy (D-LKH) protocol uses the notion of sub-trees, agreeing on a mutual key. Distributed One-way Function Tree (D-OFT) approach using logical key hierarchy in a distributed fashion was proposed by Dondeti et al, uses the one-way function tree proposed by McGrew and Sherman. Every group member is trusted with access control and key generation. In Fiat and Naor protocol, each member broadcast a single message to other participants in order to agree on a common secret.

g) Broadcast based approach

In this approach, group key is generated by broadcasting the secret messages and distributing the computations among the group members. Burmester And Desmedt Protocol is a three round protocol with member generation, broadcasting and group key computations.

Boyd proposed Conference Key Agreement (CKA) protocol, where all the group members contributed to generate the group key.

h) Comparison of Distributed Group Key Management Protocols:

All the members in the group are involved in the computation of group key or generated by one member in the group. Table III compares the distributed group key management protocols. Attributes to evaluate the efficiency of distributed key management protocols are a number of rounds, number of messages, DH key and leader requirement.

TABLE III: COMPARISON OF DISTRIBUTED GROUP KEY MANAGEMENT

Protocol	**No of Rounds**	**No of Messages**		**DH Key**	**Leader Req.**
		Uni-cast	Multi-cast		
GDH	n	n-1	n	Yes	No
Ingemarson et. al	n-1	n(n-1)	0	Yes	No
Octopus	2(n-1)/4 – 2	3n-4	0	Yes	Yes
STR	n	0	N	Yes	No
DH-LKH	$\log_2(n)$	0	$\log_2(n)$	Yes	No
D-LKH	3	N	1	No	Yes
D-OFT	$\log_2(n)$	$2\log_2(n)$	0	No	No
D-CFKM	n	2n-1	0	No	No
Fiat et. Al	2	N	N	Yes	Yes
Bermester et	3	0	2n	No	No
CKA	3	n-1	N	No	Yes

III. CURRENT RESEARCH DIRECTIONS

A group Key Management application in mobile networks, ad hoc networks, e learning, and peer-to-peer networks is prevalent. Many new protocols are proposed as existing key management protocols are no more suitable for these areas. Jiang and Hu [8] classified current group key management protocols as stateless, self-healing, distributive, reliable, adaptive and mobile-based. Among these protocols reliability and distributiveness are by default provided by the group key management protocols. Scalability of stateless group key management protocols is enhanced by reducing the degree of the polynomial functions with the help of the decentralized subgroup managers. Junbeom and Hyunsoo [12] proposed a decentralized multi-group key management scheme for stateless group members. Self-healing and rekeying are becoming target areas in the group key management protocols. Key Server transmits group key updating messages when there are some changes in membership states. A self-healing protocol can recover certain number of existing and/or future group keys. First, self-healing key distribution protocol [8] was proposed by J.Staddon et.al [25] which is based on polynomials and Angelo [10] provided an efficient self-healing scheme for LKH. Challal et al.[7] proposed adaptive group key management protocol and there is need of extensive research should be done in this

IV. CONCLUSION

This paper focused on group key management, secured distribution of session keys and refreshment of the keying material. Reviewed so many group key management protocols and placing them into three main classes: centralized, decentralized and distributed protocols, which try to minimize the requirements of KDC and group members. Centralized key management is easy to implement but more overhead on single member. The decentralized key management follows the hierarchical sub grouping and it is harder to implement. Distributed key management is simply not scalable. From the comparison tables, we analyze that no unique solution that can achieve all the requirements. Hence, it is important to understand fully the requirements of the application before selecting a security solution. A solution for secure group communication should complement a multicast application rather than drive its implementation. The usage of security mechanism for secure group communication should be made transparent to the user and it should also work well with other protocols.

REFERENCES

[1] David Manz, Jim Alves-Foss and Shanyu Zheng, “Network Simulation of Group Key Management Protocols”, Journal of Information Assurance and Security, pp. 67-79, January 2008.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.

[3] S. Rafaeli and D. Hutchison, “A Survey of Key Management for Secure Group Communication,” ACM Comput. Surv., vol. 35, no. 3, pp. 309–329, 2003.

[4] H. Harney and C. Muckenhirn. “Group Key Managemant Protocol(GKMP) Architecture”. July 1997, RFC 2093.

[5] H. Harney and C. Muckenhirn. “Group Key Managemant Protocol(GKMP) Specification”. July 1997, RFC 2094.

[6] H.H. Chu, L. Qiao, and K. Nahrstedt. A Secure Multicast Protocol with Copyright Protection. ACM SIGCOMM Computer Communications Review, 32(2):42:60, April 2002.

[7] Y. Challal, H. Bettahar, and A. Bouabdallah. “SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications”. ACM SIGCOMM Computer Communications Review, 34(2):55–70, April 2004.

[8] Bibo Jiang, Xiulin Hu, A Survey of Group Key Management, International Conference on Computer Science and Software Engineering, 2008, IEEE, DOI 10.1109/CSSE.2008.1282

[9] G. H. Chiou and W. T. Chen. Secure Broadcast using Secure Lock. IEEE Transactions on Software Engineering, 15(8):929–934, August 1989.

[10] Angelo Rossi, Samuel Pierre and Suresh Krishnan, An Efficient and Secure Self-Healing Scheme for LKH, Journal of Network and Systems Management, Vol. 18, Number 3, 327-347

[11] Debby M. Wallner, Eric J. Harder, and Ryan C. Agee. Key management for multicast: Issues and architectures. Internet draft, Network working group,september 1998, 1998.

[12] Junbeom and Hyunsoo, 2009, A decentralized multi-group key management scheme, IEICE Transactions Communications, Vol E-92-B, No.2, Feb 2009

[13] R. Canetti, T. Malkin, and K. Nissim, "Efficient Communication-Storage Tradeoffs for Multicast Encryption," in EUROCRYPT. New York, NY, USA: Springer-Verlag New York, Inc., 1999, pp. 459–474.

[14] C. K. Wong, M. Gouda, and S. S. Lam. Secure Group Communications Using Key Graphs. ACM SIGCOMM, 1998.

[15] A. Perrig, D. Song, and J.D. Tygar. ELK, A new protocol for Efficient Large-group Key distribution. IEEE Security and Priavcy Symposium, May 2001.

[16] 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 (Special Issues on Middleware), 17(8):1614–1631, August 1999.

[17] T.Hardjono, B.cain, and L.Monga. "Intra-domain Group key Management for Multicast Security". IETF internet Draft, September 2000

[18] A.Ballardie. Scalable "Multicast Key Distribution". May 1996, RFC1949.

[19] S. Rafaeli and D. Hutchison. Hydra: a decentralized group key management. 11th IEEE International WETICE: Enterprise Security Workshop, June 2002.

[20] Suvo Mittra, "Iolus: A Framework for Scalable Secure Multicasting", ACM SIGCOMM, 1997

[21] MOLVA, R. AND PANNETRAT, A. 1999. Scalable multicast security in dynamic groups. In Proceedings of the 6th ACMConference on Computer and Communications Security. (Singapore, Nov.). ACM, New York, 101–112.

[22] S.Setia, S.Koussih, S.Jaodia, and E.Harder. "Kronos: A scalable Group Re-Keying Approach for Secure Multicast". Proc. of IEEE Symposium on Security and Privacy,2000

[23] Y.R. Yang, X.S. Li, X.B. Zhang, and S.S. Lam. Reliable Group Rekeying: A Performance Analysis. TR-01-21, June 2001.

[24] KIM, Y., PERRIG, A., AND TSUDIK, G. 2000. Simple and fault-tolerant key agreement for dynamic collaborative groups. In Proceedings of the 7th ACM Conference in Computer and Communication Security, (Athens, Greece Nov.). (S. Jajodia and P. Samarati, Eds.), pp. 235–241.

[25] F.Staddon, S.Miner, M.Franklin, D.Balfanz, and D.Dean. "Selfhealing Key Distribution with Revocation". In Proc. Of the IEEE Symposium on Security and Privacy, Oakland, CA, May 2002.

[26] Alireza Nemaney Pour, Kazuya Kumekawa, Toshihiko Kato, Shuichi Itoh, "A Hierarchical group key management scheme for secure multicast increasing efficiency of key distribution in leave operation", Elsevier,Computer Networks, August 2007.

AUTHORS PROFILE

[1]R.Siva Ranjani is a research scholar in Andhra University under the supervision of Prof.P.S.Avadhani and Dr.D.Lalitha Bhaskari in Computer Science and Systems Engineering. She received her M.Tech (CSE) from Andhra University and presently working as Associate Professor in CSE Department of GMRIT. She is a Life Member of ISTE. Her research areas include Network Securiy, Cryptography, Group Key Managment.

[2]Mrs. Dr. D. Lalitha Bhaskari is an Associate Professor in the department of Computer Science and Engineering of Andhra University. She is guiding more than 8 Ph. D Scholars from various institutes. Her areas of interest include Theory of computation, Data Security, Image Processing, Data communications, Pattern Recognition. Apart from her regular academic activities she holds prestigious responsibilities like Associate Member in the Institute of Engineers, Member in IEEE, Associate Member in the Pentagram Research Foundation, Hyderabad, India.

[3]Dr. P. S. Avadhani is a Professor in the department of computer Science and Engineering of Andhra University. He has guided 7 Ph. D students, 3 students already submitted and right now he is guiding 12 Ph. D Scholars from various institutes. He has guided more than 100 M.Tech. Projects. He received many honors and he has been the member for many expert committees, member of Board of Studies for various universities, Resource person for various organizations. He has co-authored 4 books. He is a Life Member in CSI, AMTI, ISIAM, ISTE, YHAI and in the International Society on Education Technology. He is also a Member of IEEE, and a Member in AICTE.