Secure Session Key Generation Technique for Group Communication

Anurag Singh Tomar, Jaidhar C. D., and S. Tapaswi

Abstract—**In order to provide confidentiality in group communication, data is encrypted using a session key. Group key generation, session key generation and their distribution are the most important issues in multicast. This paper presents a secure session key distribution technique for group communication. In this framework, any member in the group can generate and distribute the session key to all the group members securely.**

Index Terms—**Batch rekeying, group key, multicast, session key.**

I. Introduction

In multicast sender transmits a single data packet, it is replicated and forwarded by network elements such as router/switch to the group members that are directly attached otherwise forward to the next network node. Each member in the group receives the same data. Multicast reduces the computational overhead at the sender side and utilizes the bandwidth in an efficient manner. IP multicast is used to deliver datagram to group of members simultaneously and it is useful in group oriented applications such as video conferencing, software distribution, online internet TV and audio/video streaming. These applications require confidentiality and authenticity of the group members. Authentication is the primary requirement for most of the applications and many authentication schemes have been proposed for various applications [1]. Common group key is used to provide confidentiality and authenticity among the group members. Furthermore to support dynamic membership group key needs to be updated and distributed when there is change in membership to only legitimate group members. Rekeying is used to prevent backward and forward confidentiality. Backward confidentiality means new user must not read the past communication and forward confidentiality defines ex-group member must not read the future communication. Rekeying is performed immediately whenever there is change in group membership either a single join or single leave known as immediate rekeying. Group key need to be updated whenever there is a change in the group membership which increases the communication overhead. Batch rekeying is an alternative solution to overcome this limitation. In batch rekeying [2], request for membership change are collected during time interval and processed in batch that cause change of group key. There are two kinds, a) periodic batch membership in which membership change requests are collected during period of time and processed in batch b) membership controlled batch rekeying in which requests are collected over a time period after receiving the first request and processed in a batch. Interval based rekeying maintains the rekeying frequency regardless of the dynamic change of group membership. Batched rekeying improves the performance in terms of computational overhead and bandwidth utilization. However, forward and backward access control needs slight relaxation. Critical issue in multicast is generation and distribution of a group key to only legitimate group members.

II. Related Work

Depends on the topology, group key management scheme is divided into five main classes: Centralized architecture, Broadcast architecture, Hierarchical architecture, Subgroup architecture and distributed architecture [3], [4]. In Centralized architecture, central entity is responsible for generating, distributing and updating the group key. In other words, central entity controls the whole group. The group privacy is entirely depends on the successful functioning of the single group controller. Main drawback of this scheme is failure of central entity breakdown the entire group. Furthermore, the group may become too large to be managed by a single entity thus raise the issue of scalability. Distributed architecture is one in which there is no key distribution centre and any group member can generate or distribute the group key [5]. In Hierarchical architecture, logical key tree is constructed in which leaves designated as group member, root of the logical key tree is group key and internal node is represented as Key Encryption Key (KEK). Hierarchical architecture reduces rekeying message and storage overhead at each member side. In Subgroup architecture whole group is not controlled by single centralized authority also called as decentralized architecture.

A. Group Key Management Protocol

Group Key Management Protocol (GKMP) architecture has been proposed in [6], in which each user shares Key Encryption Key (KEK) with the Group Controller (GC). Group Key Packet (GKP) generated by GC contains the Group Traffic Encryption Key (GTEK) as well as Group Key Encryption Key (GKEK). When rekey is needed, GC

Manuscript received in May 12, 2012; revised June 12, 2012. This work was supported in part by the ABV-IIITM Gwalior Department of Computer Science.

Anurag Singh Tomar is the Asst Professor at Lovely Professional University Jalandhar, India (e-mail: anuragtomar3105@gmail.com).

Jaidhar C. D. is Asst Professor at DIAT Pune, India (e-mail: cdjaidhar@rediffmail.com)

S. Tapaswi is the Professor at ABV-IIITM Gwalior, India(e-mail: stapaswi@iiitm.ac.in).

DOI: 10.7763/IJIEE.2012.V2.218

generates new GKP and encrypts it by GKEK to distribute to all the group members. New member sends a join request to GC prior to become the group member. Upon receiving a request, GC generates a GKP and encrypts it by new member KEK and sends to new member. Furthermore, it encrypts the GKP with the old GTEK and multicast the message to all the group members. When a member leaves the group, it sends the new GKP encrypted by member KEK. Hence, encryption message overhead is O (n) and GC has to store the each member KEK..

B. *Logical Key Hierarchy*

Logical Key Hierarchy (LKH) architecture has been proposed [6]-[8]. It is a tree based protocol in which GC maintains a tree of keys. Leaf node represents the member, internal node signifies the Key Encryption Key (KEK) and root node of tree is Group Key. Each group member knows the keys from leaf node to root of tree. The GC changes the group key when there is change in the membership in order to provide forward and backward secrecy. Thus, storage overhead at each member is O (log n) and communication overhead at most 2(log n).

C. *One Way Function Tree*

One Way Function Tree (OFT) has been proposed [8], [9], [10] which reduces the communication overhead from 2(log n) to (log n). Moreover, members itself computes the key by using the information provided by the GC. Each group members register its own secret key at the GC then Group Controller computes the hash of the key and sends the computed hash digest to its sibling. Upon receiving, each member also computes the hash of its own key and performs the XOR function to compute KEK. Similarly, each member and controller computes the KEK. The same procedure is repeated to compute the group key. Member computes the key from leaf node to root of tree by the following equation $K_i=f(g(k_{left(i)}),g(k_{right(i)}))$ where left(i) and right(i) denotes left and right children of node i respectively, f is the XOR function and g is the one way hash function and K_i is the KEK.

D. *Iolus*

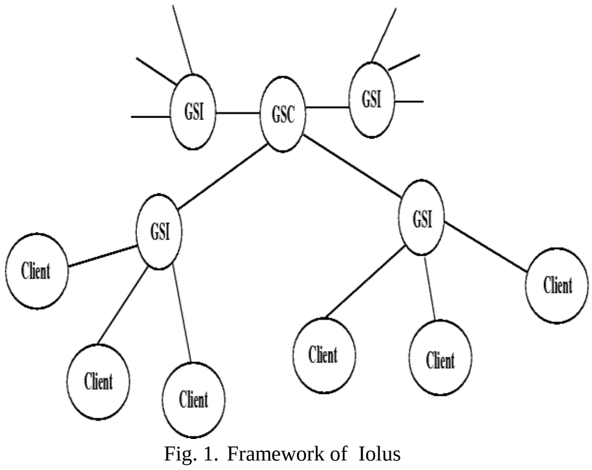


Fig. 1. Framework of Iolus

Iolus Architecture has been proposed in [11], [12] for group key management and the same is shown in Fig. 1. It divides the whole group into subgroups; each subgroup is controlled by Group Security Agent (GSA). These GSA form another group that is controlled by another group GSA. Root of the tree called Group Security Controller (GSC) and others called as Group Security Intermediaries (GSI). Each subgroup holds subgroup key which is different from other subgroup key. Members of the top level group called as GSI shares a key for the group. When a subgroup member wish to communicate to whole group, it encrypt the data by subgroup key and multicast the encrypted data to the subgroup. Upon receiving the encrypted data, GSA decrypts the received data and again encrypts it by top level group key. It sends to top level group so each of GSA receive the encrypted data and decrypt it and further encrypt it with its own subgroup key and send to all the members. When there is change in membership, changes do not affect the entire group, only the subgroup where there is change affected. However, main disadvantage is more computation and that data translation.

III. Proposed Scheme

In this paper decentralized group key distribution technique with batch rekey is proposed. Any member of group U_i who wants to initiate the session can generate the session key SK and multicast the information to entire group so that each member derive the session key SK from the received information. In addition, they can verify whether the derived session key is correct or not. In this proposed technique, users generate one long term secret 'K' using Tree Based Group Diffie-Hellman (TGDH) and are as shown in Fig. 2. Only group members know the long term secret. Table I shows the steps of generation, distribution, derivation and verification of derived session key. The notations used in this paper are defined as follows.

$H()$	:Secure One way hash function
p	:Prime Number
r_1, n_1	:Random Numbers
SK	:Session Key
$E_{H(k)}[T]$	:Message T is encrypted by key H(K)
$D_{H(k)}[T]$	:Message T is decrypted by key H(K)
g	:Generator
K	:Long term secret

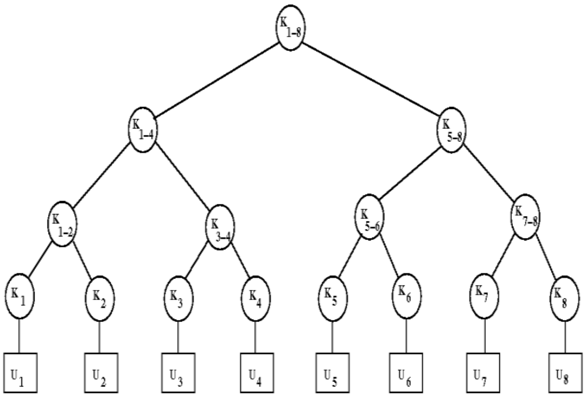
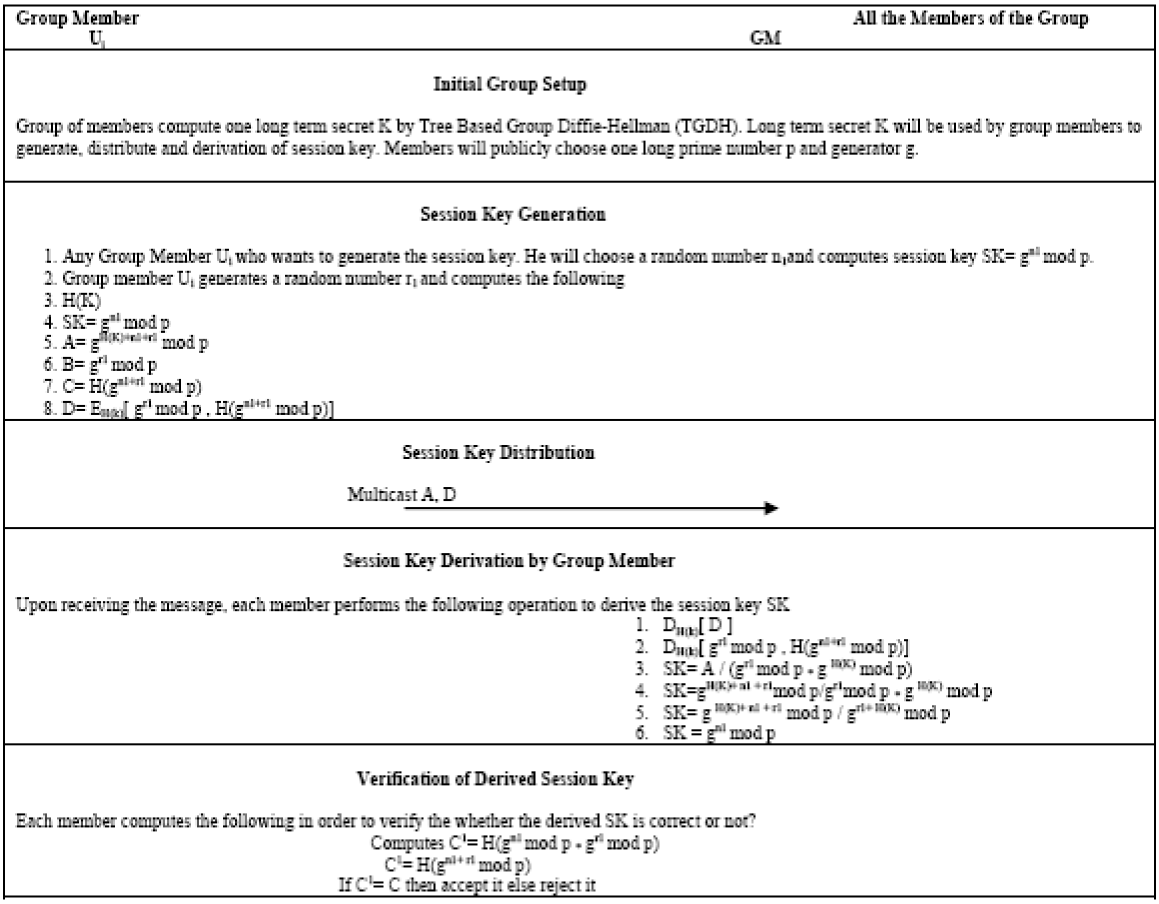


Fig. 2. Tree based group diffie-hellman

IV. Security Analysis

User U_i selects a random number to compute the session key for each session, so, they are independent for every session. The security of proposed scheme depends on the group key K which is known only to the group members. To derive the session key from the intercepted message A= $g^{H(K)+n1+r1}$ mod p, D=$E_{H(k)}$[g^{r1} mod p , H(g^{n1+r1} mod p)], attacker needs group key to decrypt.



Group Member U_i **GM** **All the Members of the Group**

Initial Group Setup

Group of members compute one long term secret K by Tree Based Group Diffie-Hellman (TGDH). Long term secret K will be used by group members to generate, distribute and derivation of session key. Members will publicly choose one long prime number p and generator g.

Session Key Generation

1. Any Group Member U_i who wants to generate the session key. He will choose a random number $n_1$and computes session key SK= g^{n1} mod p.
2. Group member U_i generates a random number r_1 and computes the following
3. H(K)
4. SK= g^{n1} mod p
5. A= $g^{H(K)+n1+r1}$ mod p
6. B= g^{r1} mod p
7. C= H(g^{n1+r1} mod p)
8. D= $E_{H(k)}$[g^{r1} mod p , H(g^{n1+r1} mod p)]

Session Key Distribution

Multicast A, D →

Session Key Derivation by Group Member

Upon receiving the message, each member performs the following operation to derive the session key SK

1. $D_{H(k)}$[D]
2. $D_{H(k)}$[g^{r1} mod p , H(g^{n1+r1} mod p)]
3. SK= A / (g^{r1} mod p ∗ $g^{H(K)}$ mod p)
4. SK=$g^{H(K)+n1+r1}$mod p/g^{r1}mod p ∗ $g^{H(K)}$ mod p
5. SK= $g^{H(K)+n1+r1}$ mod p / $g^{r1+H(K)}$ mod p
6. SK = g^{n1} mod p

Verification of Derived Session Key

Each member computes the following in order to verify the whether the derived SK is correct or not?

Computes C^1= H(g^{n1} mod p ∗ g^{r1} mod p)

C^1= H(g^{n1+r1} mod p)

If C^1= C then accept it else reject it

Proposed scheme provides perfect forward secrecy. One session key does not supply any information to derive past and future session keys. Even if an attacker successful to obtain one session key does not provide any information about present and future communication because session keys are independent of each other.

New member not be able to access group data until the join request gets accepted. Membership join/leave requests collected over a period of time are executed in a batch. Group key is updated after processing the join/leave request. Hence, member who left the group cannot decrypt the group data by previous session key.

In every session, different random number is used to compute the session key. Information obtained in session cannot be used again and again. Hence, scheme is secure against the replay attack.

V. Conclusion

Most important issue in multicast is key generation and its distribution because group membership is dynamic in nature. Secure session key distribution technique is proposed in this paper. Any legitimate group member U_i can generate and distribute the session key to all the group members.

References

[1] J. Choi, S. Jung, K. Bae, and H. Moon, “A Lightweight Authentication and Hop-by Hop Security Mechanism for SIP Network,” *International Conference on Advanced Technologies for Communications*, 2008.

[2] T. Aurisch, “Using Key Trees For Securing Military Multicast Communication,” *IEEE MILCOM*, 2004.

[3] S. Q. Li and Y. Wu, “A Survey on Key Management for Multicast,” *Second International Conference on Information Technology and Computer Science*, 2010.

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

[5] B. Jiang and X. Hu, “A Survey of Group Key Management,” *International Conference on Computer Science and Software Engineering*, 2008.

[6] B. Jiang and X. Hu, "A Survey of Group Key Management," Computer Science and Software Engineering, 2008 International Conference on , vol. 3, no. 12-14, pp. 994-1002, 2008, ,doi: 10.1109/CSSE.2008.1282. A Survey of Group Key Management [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4722510& isnumber=4722268

[7] H. Hugh and E. Harder, “Logical Key Hierarchy Protocol,” *Internet Draft, Draft-harney-sperta-lkhp-sec-oo.txt, Internet Engineering Task Force*, March 1999.

[8] S. M. Ghanem and H. A. Wahab, “A Secure Group Key Management Framework: Design and Rekey Issues,” in *Proceedings of the Eighth IEEE International Symposium on Computers and Communication* 2003.

[9] A. T. Sherman and D. A. Mcgrew, “Key Establishment In Large Dynamic Groups Using One-Way Function Trees,” *IEEE Transactions On Software Engineering*, vol. 29, no. 5, May 2003.

[10] S. Xu, Z. Yang, Y. Tan, W. Liu, and S. Sesay, “An Efficient Batch Rekeying Scheme Based on One-Way Function Tree,” in *Proceedings of ISCIT*, 2005

[11] S. Mittra, "Iolus : A Framework for Scalable Secure Multicasting," *ACM SIGCOMM*, 1997

[12] M. Peyravian, S. M. Matyas, and N. Zunic, "Decentralized group key management for secure multicast communications," *Computer Communications*, pp.1183–1187, 1999.

Anurag Singh Tomar is currently Asst. Professor at Lovely Professional University Jalandhar, India. He received Master of Technology From ABV-IIITM Gwalior, India in Information Security Specialization.

Jaidhar C.D. received the Ph.D**.** in Computer Science Engineering. He is currently Asst. Professor at Defence Institute of Advanced Tehnology Pune, India. His research interest includes Network Security, Smart card Authentication, Cyber Crime, Computer Networks and Quality of Service(QOS).

S. Tapaswi received the Ph.D**.** in Computer Science Engineering. She is currently Professor at ABV-IIITM Gwalior, India. Her Area of Research Includes Image Processing, Mobile Ad-hoc Networks and wireless communication.