MINIMISATION OF KEY GENERATION OVERHEAD IN GROUP KEY MANAGEMENT WITH MODULAR EXPONENTIAL FUNCTION APPROACH

Rakesh Kumar[1], Akhilesh Kumar[2], Samir Shrivastava[3] and Ashish Kumar Shrivastava[4]

[1,2,4]IT Department REC, Ambedkar Nagar
[3]CSE Department KNIT, Sultanpur.

ABSTRACT

Multicast communication experiences recipient get to issue due to forward secrecy, backward secrecy. The group key management is a productive component to handle this circumstance. Be that as it may, there are numerous entities which impact the communication, computation overhead, message size, storage overhead and so forth. There are we require proficient group key management path to deal with secure the frame function and decrease the overhead in the current approach by utilizing Key Graph. Exist key graph [7] proposed the augmentation of the double key tree to 4-ary key tree. 4-ary key tree beat the issue of re-entering as far as height of the key tree. Using a greater degree reduces the height of the key tree and, as a result, improves re-keying performance with the help of Modular Function.

KEYWORDS

Key Management, Key Generation, Modular Function, Group key management, Multicast Security.

1. INTRODUCTION

Computer system is one of the important members of current period. A Computer system is essentially the collection of computers and different kind of gadgets that are interfaced by different resources i.e. communications channels that give the communications among clients and permits them of sharing the resources [8]. Multicast is one of the service that provide different type of paths for communication such as one to many, many to one, many to many. Multicast refers to the transmission of a message from one sender to multiple receivers or from multiple senders to multiple receivers [1]. There are four types of multicast security such as multicast receiver access control, multicast source authentication, multicast fingerprinting and group key management. All these multicast security have a few issues and researchers gave solution for multicast security issues.

Multicast Security

There are four types of multicast security [4,13,15] such as multicast receiver access control, multicast source authentication, multicast fingerprinting and group key management.

The routing protocols must be aware of group members in the system in order to deliver packets to them. The mechanism provided for doing this is the Internet Group Membership Protocol (IGMP). A host uses this protocol to notify the routing system that it should deliver packets for a particular multicast group to this host. In the current model, any host can use IGMP to become a member of any IP multicast group, causing eavesdropping or theft of service. The traditional method used to

protect the information is to encrypt the multicast information and provide decryption keys only to authorized members. Different type of issues in Multicast receiver Access Control on the basis of required functionality and Components are given below-

Required Functionality: The functions necessary to provide controlled access to a group are as follows-

Group Policy Specification Functions: - The group policy is an access control policy that specifies among other things which hosts have access rights to become members. The group Controller is the entity that has been assigned membership of the multicast group and is allowed to specify the group policy.

Access Request Functions: - Access request function provides information about a group member and it belongs from a certain group.

Access Control Functions: - These involve receiving a host's request, authenticating the host and performing authorization. Authorization requires checking the group policy to determine if that host has the access rights to become a member of the requested group [3, 6].

Components: - Multicast receiver access control architecture is composed two types of systems one of the Group Policy Management System and second one of the Group Member Authorization System. Multicast receiver access control architecture also interacts with the routing system and any group key management system that are given blows.

Group Policy Management System: - The group policy management system involves a group owner providing the list of authorized members and possibly other security policy for the group to the access control server (ACS).

Group Member Authorization System: - The group member authorization system provides the core functionality of group access control architecture by controlling access to the group. The design goals of an authorization system are to maintain security and achieve scalability [6].

Multicast Receiver Access Control: Solving these problems which are discussed in previously requires controlling the ability of hosts to join the multicast group [14]. Author calls this multicast receiver access control. The need for a solution to these problems is well known. The term secure IGMP has been used to refer to the protocol that would provide the solution.

Multicast receiver authorization solutions can be classified based on how they provide revocations. Some systems do not provide revocation, some systems leverage the authorization state maintained by some outside system, some systems must query a centralized server to maintain authorization state, other systems distribute access control lists to routers, and some systems efficiently provide revocation using time-limited authorizations.

Multicast Source Authentication

In Source authentication is the ability of group members to verify the identity of the sender of a received packet [3]. In unicast, a shared secret key message authentication code (MAC) is used to provide authentication. In multicast, the group key provides a shared secret key; however, performing message authentication with this key only verifies that the sender is a member of the group, but not necessarily the intended source. Many applications require a level of authentication that allows a receiver to identify the member sender of a message. There has been function that aims to efficiently provide this level of source authentication. Source authentication faces different

type of issues in the multicast communication such as Authenticity, Integrity, Non-repudiation, Efficiency, Collision Resistance, Minimal Latency, and Robustness against unreliable communication.

Solutions for Multicast Source Authentication: There are two mechanisms for the multicast source authentication schemes such as hash-based mechanisms and MAC based mechanisms [3, 5, 9].

Proposed Function: Multicast communication suffers from receiver access problem due to forward secrecy, backward secrecy. The group key management is an efficient mechanism to handle this situation. But there are many factors which effect the communication, computation overhead, message size, storage overhead, these factors are as following:

Heterogeneous nature of the group membership affects the possible type of encryption algorithm to be used, and the length of the key that can be supported by an end user.

The cost of setting up and initializing the entire system parameters, such as selection of the group controller (GC), group announcement, member join and initial key distribution.

Administrative policies, such as those defining which members have the authorization to generate keys.

- Required level of performance of parameters such as session sustainability, and key generation rates.

- Required additional external support mechanisms, such as the availability of a certificate authority (CA).

There are we require efficient group key management approach to secure the system and reduce the overhead in the existing approach [2]. Exist key graph [7] proposed the extension of the binary key tree to 4-ary key tree. 4-ary key tree overcome the problem of re-keying in terms of height of the key tree. Using a greater degree reduces the height of the key tree and, as a result, improves re-keying performance. Performance of re-keying measured in terms of computation overhead, communication overhead, message size and storage overhead. Really, optimal results are gained when the tree has a degree of 4. In the figure 1(a) illustrates the logical key tree with two nodes when there are seven joining members (u_1 through u_7). When u_8 joins, the key server first attaches it to node $K_{1,2}$ as shown in figure 1(b), and then, changes the group key K_G and the node key $K_{1,2}$ to $K`_G$ and $K`_{1,2}$ respectively. For delivering them, each new key is encrypted with the previous one (K_G and $K_{1,2}$ respectively), and a set of them are sent by multicast for existing members. For the new member they are sent by unicast being encrypted with its session key. On the other hand, when a member leaves the group, new keys are encrypted by their corresponding child keys, and a set of them are sent for remaining members by multicast. For example, when member u8 leaves the group shown in figure 1(b), the key server changes $K`_{1,2}$ and $K`_G$ to $K``_{1,2}$ and $K``_G$ respectively. Then, it delivers $K``_{1,2}$ for $\{u_5,u_6,u_7\}$ being encrypted by $K_5$, $K_6$ and $K_7$, and $K``_G$ for $\{u_1,u_2,u_3,u_4\}$ and $\{u_5,u_6,u_7\}$ being encrypted by $K_{1,1}$ and $K``_{1,2}$ respectively. A set of these keys are sent by one multicast message.

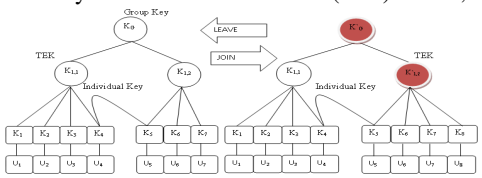

Figure 1: Logical Key Tree for Key Graph

Proposed Protocol

The proposed protocol depends on key graph that deals with the entire group on the premise of logical 4-ary key tree or key tree is the extended version of binary tree. In this protocol, we have separated entire group in a few subgroups and subgroup organized in a logical key hierarchy as in 4-ary key tree which diminish the complexity for a member join or leave from O (m) to O ($\log_4$ n/m). The members in every subgroup contribute with each other to create the subgroup key. This procedure appoints the key redesign prepare at a leave Procedure from the key server side to the member side. The proposed protocol functions in a hierarchy of two levels of controllers; the first for the group controller (GC) and the second is the subgroup controller (SC). The GC shares a symmetric key to all SCs which are trusted entitys. The member of the SCs is to make an interpretation of the information going to their subgroups. Every SC acts as the server of its subgroup. The fundamental goal of this protocol is to management a symmetric key between all group members keeping in mind the end goal to preserve the security of group communication. If there should arise an occurrence of dynamicity happens in the group membership by joining or leaving the group, the group key should be updated to keep up in backward secrecy and forward secrecy.

The subgroup is organized in a hierarchy like the LKH approach [11] and KS is the key of the group key. For the Process of the proposed protocol are following:

In this approach key server is a trusted entity which in- charge of create required keys and for appropriating those keys to legitimate group members and Each member from the group has IGMP membership, when new member joins a group; it sends an IGMP membership report message to its neighboring switch to have the multicast information conveyed from a multicast sender. Other side, the member sends a join request for message to the key server to acquire the group key by which the multicast information is encrypted. This is not quite the same as other LKH approaches, in term to handle a large number of members effectively; our approach divides group members into subgroups. For example 256 members are divided into 16 subgroups.

Our approach applies the concept of key tree in LKH to the subgroups. In the logical key tree, leaf nodes correspond to subgroups, not individual members. Like other LKH approaches, the root node relates to the group key, and the middle nodes compare to traffic encryption key (TEK) transfer for key exchanges.

The division of group members into subgroups is performed so that a balanced tree is developed. For this situation, by dividing n (256) members into subgroups whose size is m (16) members, we will have Γn/mד subgroups, and the height of the tree will be $\log_4 \lceil n/m \rceil$. For example the group divided into 16 subgroups from 1 to 16 subgroups and height of the tree is 3. At the point when a member joins a group, it is allocated to a subgroup. At this time, the member obtains the following three kind of key information from the key server.

The private key: This key which is shared just to the key server is transfer to exchange information safely between the key server and the member. In addition, this key might be a predetermined key, or assigned dynamically through a safe channel, for example, TLS [10] The keys which are characterized as per the logical key tree: As described above, they incorporate the subgroup key, the node keys and the group key, along the path from the leaf node (comparing to the subgroup) to the root node. We call these keys the path set.

The key information of different members in the subgroup: This information is used to change the subgroup key, the node keys and the group key when a member in the subgroup leaves the multicast group. This information is called the inverse value in the rest of the paper. The subtle elements of these keys are described below in this area and in the following segment. For producing subgroup keys, node keys and group keys, our approach uses the modular exponential calculation over the finite field [9] using member secrets and the server secrets. Member secret is a value which is assigned dynamically to every member at join and server secret is a value which is assigned dynamically to every subgroup at join to ensure backward secrecy. These values are generated by the server at join (see next section). Moreover, each subgroup key is generated by using the member secrets and the server secret assigned to that subgroup with modular exponentiation. Next, the node keys and the group key are calculated from two child node keys in the logical key tree.

Key Generation

As said above, we use the modular exponential function as a one-way function. Since p is a large prime and g is the primitive element of multiplicative group Z_p^* it is computationally hard to determine α given g and gα (mod p). Based on this property, the subgroup keys, the node keys and the group key are organized as follows.

First of all, the member secret is selected under the condition that $2 \leq \alpha_j^s \leq p$ and gcd (,p-1)=1.

The server secret the server secret is selected under the condition that is selected under the condition that $2 \leq \alpha_j^s \leq p$.

Using those secrets, the subgroup key for subgroup j is calculated by $K_j \equiv g\alpha_j^1\alpha_j^i \ldots\ldots\ldots\ldots \alpha_j^m \pmod p$. The node keys and the group key are organized by multiplying the exponents of its two child node keys (or the subgroup keys) in the logical key tree.

In order to illustrate the algorithm for re-keying, we use a simple example of multicast group divided into 16 subgroups; subgroup 1 to 16 with m members and subgroup 16 with m -1 members respectively. Figure.2 depicts the logical key tree for this group. The members of subgroups 1,2,3,4 own subgroup keys K_1, K_2, K_3 and K_4 respectively, node key $K_{1,4}$. The members of subgroups 5,6,7,8 own subgroup keys K_5, K_6, K_7 and K8 respectively, node key $K_{5,8}$. The members of subgroups 9,10,11,12 own subgroup keys K_9, K_{10}, K_{11} and K_{12} respectively, node key $K_{9,12}$. The members of subgroups 13,14,15,16 own subgroup keys K_{13}, K_{14}, K_{15} and K_{16} respectively, node key $K_{13,16}$ and group key K_G. In this process key server used pre- computational function (PK) for calculating key when member join or leave the group and by using this pre-computational function process , we have minimized the computational cost during key generation and the keys are calculated as follows:

$$K_1 \equiv g\,\alpha_1^1 \ldots\ldots \alpha_1^{m-1}\alpha_1^m \pmod p$$
$$K_2 \equiv g\,\alpha_2^1 \ldots\ldots \alpha_2^{m-1}\alpha_2^m \pmod p$$

$K_1 \equiv g^{\alpha_1^1 \ldots \alpha_1^{m-1} \alpha_1^m} \pmod p$
$K_2 \equiv g^{\alpha_2^1 \ldots \alpha_2^{m-1} \alpha_2^m} \pmod p$
$K_3 \equiv g^{\alpha_3^1 \ldots \alpha_3^{m-1} \alpha_3^m} \pmod p$
$K_4 \equiv g^{\alpha_4^1 \ldots \alpha_4^{m-1} \alpha_4^m} \pmod p$
$K_5 \equiv g^{\alpha_5^1 \ldots \alpha_5^{m-1} \alpha_5^m} \pmod p$
$K_6 \equiv g^{\alpha_6^1 \ldots \alpha_6^{m-1} \alpha_6^m} \pmod p$
$K_7 \equiv g^{\alpha_7^1 \ldots \alpha_7^{m-1} \alpha_7^m} \pmod p$
$K_8 \equiv g^{\alpha_8^1 \ldots \alpha_8^{m-1} \alpha_8^m} \pmod p$
$K_9 \equiv g^{\alpha_9^1 \ldots \alpha_9^{m-1} \alpha_9^m} \pmod p$
$K_{10} \equiv g^{\alpha_{10}^1 \ldots \alpha_{10}^{m-1} \alpha_{10}^m} \pmod p$
$K_{11} \equiv g^{\alpha_{11}^1 \ldots \alpha_{11}^{m-1} \alpha_{11}^m} \pmod p$
$K_{12} \equiv g^{\alpha_{12}^1 \ldots \alpha_{12}^{m-1} \alpha_{12}^m} \pmod p$
$K_{13} \equiv g^{\alpha_{13}^1 \ldots \alpha_{13}^{m-1} \alpha_{13}^m} \pmod p$

Join Process

We now use to explain how re-keying is done when a new member joins the multicast group. In this process key server used pre-computational function for calculating key when member join or leave the group and this pre-computational function process minimized the computational cost during key generation. The procedure is as follows:

When key server receives a join request, it authenticates the member. This may be done by the protocolal approach such as remote authentication dial in user service (RADIUS) [56], and we do not discuss this procedure. If required, the key server assigns the session key, and sends it to the member.

The key server determines the subgroup for the new member and assigns the identity within the subgroup. In this example, the new member belongs to subgroup 16 and its identity is m. At this time, the path set for subgroup16, the keys K_{13}, K_{14}, K_{15}, K_{16} and K_G need to be changed to new ones.

The key server assigns member secret to , and calculates its inverse value as well. The key server changes the server secret assigned to subgroup 16 from to .

The key server updates K_{16}, $K_{13,16}$ and K_G to $K`_{16}$, and $K`_G$ using and in the following path.

$K`_{16} \equiv g^{\alpha_{16}^1 \ldots \alpha_{16}^{m-1} \alpha_{16}^m} \pmod p$
$K`_{13,16} \equiv g(\pi_i \alpha_{13})(\pi_i \alpha_{14})(\pi_i \alpha_{15})(\pi_i a \pmod p$
$K`_G \equiv (PK_{13,16} PK_{9,12} PK_{5,8})^{\alpha_{16}^1 \ldots \alpha_{16}^{m-1}} \alpha_{16}^m a \pmod p$

The key server encrypts $\{K`_{16}, K`_{13,16}, K`_G\}$,and the inverse values of the other members in that subgroup, …… by K than it sends this encrypted message through unicast to It has been given below:

S { }:{($K`_{16}$, $K`_{13,16}$, $K`_G$, ……) K }.

Server encrypts , and $K`_{16}$ by $K_{16}$ for subgroup 16, $K`_{13,16}$ by $K_{13,16}$ for subgroup 13,14,15, $K`_G$ by K_G for subgroup 1 to 12,and distributes these encrypted keys through multicast for existing members. This process describe as following:

S —Multi→ {Existing Members}:{(, K`16) K16,( K`13,16) K13,16 ,(K`G) KG }. And the complete process of join a group and assign the key is give in figure 2.

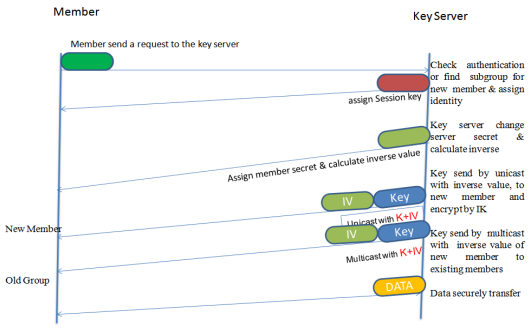

Figure 2: Joining Process of Proposed Approach

In this process, each updated key is encrypted by the previous one for existing members, and as a result, only the members who know the corresponding previous keys can decrypt the encrypted message containing the new keys.

Leave Process

When user leaves the group then all member of the group affected by this change and key server changes the group key or path key such as K`16 to K``16, K13,16 to K``13,16 and KG to K``G. According to our protocol, these updated keys do not need to be sent to the remaining members. Instead, the key server just prepares one message for subgroup 16 indicating leaves and delivers for subgroup 1 to 15. The value of is encrypted into multiple copies by K15 and K13,16, for subgroup 15 and 1 to 14 respectively. The key server sends this message through multicast. This process describe as following:

S —Multi→ {Remaining members}:{( ) K15,( ) K13,16}.

And the complete process of join a group and assign the key is give in figure 3.

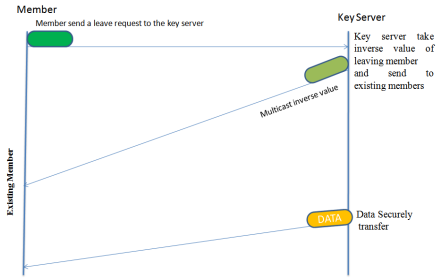

Figure 3: Leaving Process of Proposed Approach

When the remaining members receive this message, they decrypt it by the corresponding keys and then Use to update those keys.

$$K``_{16} \equiv (K`_{16}) \circ (\text{mod } p) \equiv g \quad \ldots\ldots \quad \propto_{16}^{-m} \propto_{16}^{m} \quad (\text{mod } p)$$
$$K``_{16} \equiv g \quad \ldots\ldots \quad \propto_{16}^{m} \propto_{16}^{-m} \quad (\text{mod } p)$$

$$K``_{13,16} \equiv (K`_{16}) \circ \quad (\text{mod } p)$$
$$\equiv g\,(\pi_i)\,(\pi_i)(\pi_i)(\pi_i) \quad (\text{mod } p)$$
$$\equiv g\,(\pi_i)\,(\pi_i)(\pi_i)(\pi_{i-1}) \quad (\text{mod } p)$$

$$K``_{G} \equiv (K`_{G}) \circ \quad (\text{mod } p)$$
$$\equiv (PK_{13,16}\, PK_{9,12}\, PK_{5,8}) \quad \ldots\ldots \quad \propto_{16}^{-m} \propto_{16}^{m} \quad (\text{mod } p)$$
$$\equiv (PK_{13,16}\, PK_{9,12}\, PK_{5,8}) \quad \ldots\ldots \quad \propto_{16}^{-m} \propto_{16}^{m} \quad (\text{mod } p)$$

As we notice, the key server does not need to generate new keys (TEK and group key) after a leave. Instead, it just sends the inverse value of leaving member to remaining members. Then, the remaining members update the necessary keys. In this path, updating the keys after a leave is shifted to member's side which improves the efficiency of re-keying at leave.

Simulation Environment

We are using QualNet simulator version 5.0 to simulate our function. QualNet simulator provides wide variety of simulation platform that can predict wireless, wired and mixed platform net function and net functioning device performance. QualNet software can explore and analyze early-stage device designs and application code in closed, synthetic net functions at real time speed or faster. QualNet allows users to set up, develop, and run custom system models. A feature-rich visual development environment allows users to set up models quickly, efficiently code protocols, and then run models that present real-time statistics and helpful packet-level debugging insight. QualNet supports over thousands of net function nodes [12].

Key Generation Overhead:

Key generation overhead at the key server and member node along the path to the root at each join or leave Process and formula for key generation overhead summarize in table 2. as following:

Table 1: Key Generation

Approaches	Process on Key Server Node		Process on Member Node	
	Join	Leave	Join	Leave
Simple Application	2	1	0	0
LKH	n	n	0	0
OFT	n	n-1	n	n
Our Protocol	$\log_4 \lceil n/m \rceil$	0	0	0

Table 1 shows that the number of key generations at the key server is almost equal to the height of the key tree. First of all, Simple App. has the smallest overhead at the key server both join and leave process. Our approach minimizes number of key generation at the key server both join and leave process as compare to LKH and OFT. By contrast, because of smaller size of hsg, the key server generates fewer keys at join. Most importantly, the key server does not need to generate new keys for the members at leave.

On the other hand in simple application and LKH, a member node does not generate any keys by it at each event, but in OFT the new member at join and a remaining member node along the path at leave need generate new node keys by mixing two hash values. At a member leave process the group and subgroup controller doesn't generate any keys. Instead it multicasts the identity of the leaving member to all the group and subgroup members to be factored from the subgroup key by using the leaving member's inverse value. Figure 4(a) and 4(b) shows comparative result of number of key generation overhead on the basis of group size and number of key generated at the key server.

From the figure 4(a) one can notice that the proposed protocol has minimized overhead at the join process because the key server reduced the height of key tree by using 4-ary key tree.

From the figure 4(b) one can notice that the proposed protocol has the smallest overhead at the leave process because the key server doesn't generate any keys in that case.

Figure 4(a): Key Generation Overhead at the Key Server during Join Process

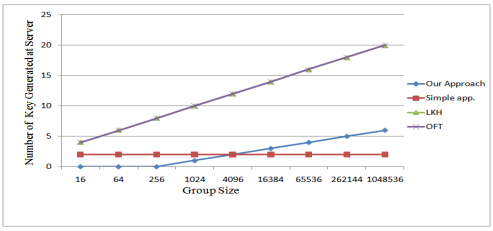

Figure 4(b): Key Generation Overhead at the Key Server during Leave Process

Figure 5(a) and 5(b) shows comparative result of number of key generation overhead on the basis of group size and number of key generated at the member node.

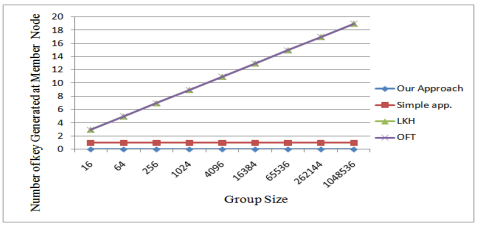

Figure 5(a): Key Generation Overhead at the Member Node during Join Process

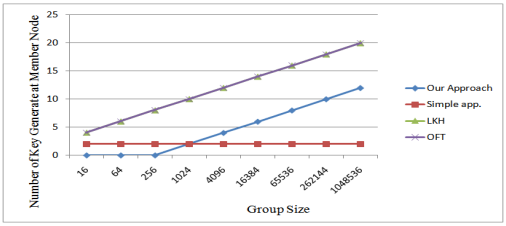

Figure 5(b): Key Generation Overhead at the Member Node during Leave Process

The key generation overhead for our protocol is 0 at join, but proportional to the height of the key tree at leave as shown in the table 1 at member node. In fact, a member node renews the node keys along the path to the root by modular exponentiation.

3. CONCLUSIONS

In this paper, we have proposed a security improvement in group key management approach to solve the problem of distributing a symmetric key between the whole group members for secure group communication. Our approach divides a group of n members into subgroups and size of each subgroup is m members. Each subgroup assign to leaf node of the key tree. When n>2m, we have subgroups and height of the logical key tree will be. .

We are using inverse values and pre-computation function. The purpose of inverse value is to assign member secret to each member, in order to update the keys efficiently at leave, the keys calculated by the members rather than delivered by key server. Pre-computation function PK will minimize the computation cost during join or leave process. After join or leave process key server re-calculates the key and that key delivered to the members, during re-keying calculation pre-computation function improve the performance of the key server.

REFERENCES

[1] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. Chidambaram Kalimuthu, and R. Dharmaraj" Secure Group Key Management Scheme for Multicast Systems", International Journal of System Security, Vol.11, No.1, PP.33-38, July 2010.

[2] Imane Aly Saroit, Said Fathy El-Zoghdy, and Mostafa Matar," A Scalable and Distributed Security Protocol for Multicast Communications", International Journal of System Security, Vol.12, No.2, PP.61-74, Mar. 2011.

[3] Paul Judge and Mostafa Ammar," Security Issues and Solutions in Multicast Content Distribution: A Survey", IEEE System, January/February 2003.

[4] Xirong Bao,Jin Liu, Lihuang She, Shi Zhang "A key Management Scheme Based on Grouping within Cluster", Intelligent Control and Automation (WCICA), 2014 11th World Congress on 29 June-4 July 2014.

[5] William Stallings" Cryptography and System Security Principles and Practices," Fourth Edition, Pages: 592, November 16, 2005.

[6] P. Q. Judge and M. H. Ammar, "Gothic: Group Access Control Architecture for Secure Multicast and Anycast," IEEE INFOCOM, July 2002.

[7] C.K. Wong, Mohamed Gouda, and Simon S. Lam, "Secure group communications using key graphs", IEEE/ACM Transactions on Systeming. 8 (1) (February 2000) 16–30.

[8] A. Tanenbaum, Computer Systems, Fourth Edition, Prentice Hall, 2009

[9] Douglas R. Stinson, "Cryptography Theory and Practice", Third edition, Chapman and Hall/CRC Press, 2002, pp. 119–155.

[10] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS)" Protocol Version 1.1, RFC2346 (April 2006).

[11] D. Wallner, E. Harder, R. Agee", Key Management for Multicast: Issues and architectures", National Security Agency, RFC2627 (June 1999).

[12] http://www.scalable- systems.com/ content/products /qualnet.

[13] Wee Hock Desmond Ng, Michael Howarth, Zhili Sun, and Haitham Cruickshank," Dynamic Balanced Key Tree Management for Secure Multicast Communications," IEEE Transactions on Computers, VOL. 56, Page 590 - 605, MAY 2007.

[14] Haibin Lu," A Novel High-Order Tree for Secure Multicast Key Management," IEEE Transactions on Computers, VOL. 54, Pages: 214 - 224, February 2005.

[15] D. Cheriton and S. Deering, "Host Groups: A Multicast Extension for Datagram Internetworks," Information Communication. Symp., Sept. 1985, pp. 172-79.

AUTHORS

Akhilesh Kumar graduated from Mahatma Ghandhi Mission's college of Engg. and technology, Noida, Uttar Pradesh in Computer Science & Engineering in 2010. He has been M.Tech in the department of Computer Science & Engineering, Kamla Nehru Institute of Technology, Sultanpur (Uttar Pradesh). SinceAugust2012, he has been with the Department of Department of Information Technology, Rajkiya EngineeringCollege, Ambedkar Nagar, as an Assistant Professor.His area of interests includes Computer Networks and Mobile ad-hoc Nerwork.

Ashish kumar shrivastava is lecturer in Computer Science &Engineering Department and become member of different committeesACS, Paper setter, External Examiner And since 2012, he has been with the Department of Information Technology, Rajkiya Engineering college, Ambedkarnagar (UP) as an Assistant professor and hold various post like Dean Academic affairs, Center superintendent, Member of Proctorial board etc. His current research areas are in Biometrics system, Multicast security, Face recognition system, Bio-Medical Multimodal.

Rakesh Kumar was born in Bulandshahr (U.P.), India, in 1984. He received the B.Tech. degree in Information Technology from Kamla Nehru Institute of Technology, Sultanpur (U.P.), India, in 2007, and the M.Tech. degrees in ICT Specialization with Software Engineering from the Gautam Buddha University, Greater Noida, Gautam Budh Nagar, Uttar Pradesh, India, in 2012.In 2007, he joined the Quantum Technology, New Delhi as a Software Engineer and Since August 2012, he has been with the Department of Department of Information Technology, Rajkiya Engineering College, Ambedkar Nagar, as an Assistant Professor. His current research interests include Computer Network, Multicast Security, Sensor Network and data mining. He is a Life Member of the Indian Society for Technical Education (ISTE), and he is a Nominee Member of Computer society of India.