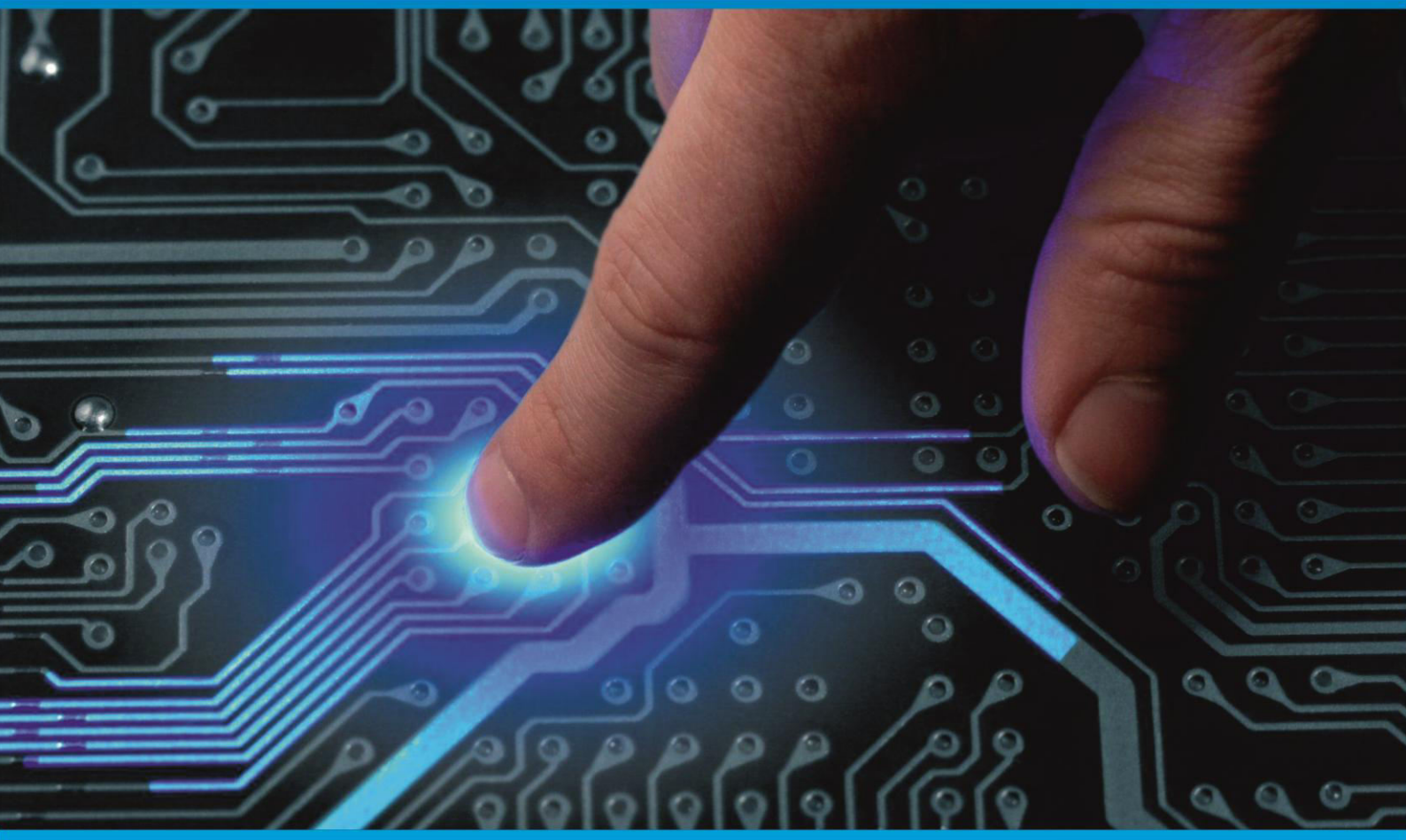
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 4, April 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

Group Key Management Protocol

Sundar E, Naveen S, Prabhat B, Dr.J.Jospin Jeya M.E,Ph.d.,

Department of Computer Science and Engineering, Jeppiaar Engineering College, Chennai, Tamilnadu, India

Department of Computer Science and Engineering, Jeppiaar Engineering College, Chennai, Tamilnadu, India

Department of Computer Science and Engineering, Jeppiaar Engineering College, Chennai, Tamilnadu, India

Associate Professor, Department of Computer Science and Engineering, Jeppiaar Engineering College, Chennai, Tamilnadu, India

ABSTRACT: Group key management is an important functional building block for any secure multicast architecture. Thereby, it has been extensively studied in the literature. In this paper we present relevant group key management protocols. Then, we compare them against some pertinent performance criteria.

KEYWORDS: Multicast, Security, Group Key Management.

I. INTRODUCTION

THE phenomenal growth of the Internet in the last few years and the increase of bandwidth in today's net works have provided both inspiration and motivation for the development of new services, combining voice, video and text "over IP". Although unicast communications have been predominant so far, the demand for multicast communications is increasing both from the Internet Service Providers (ISPs) and from content or media providers and distributors.

Indeed, multicasting is increasingly used as an efficient communication mechanism for group-oriented applications in the Internet such as video conferencing, interactive group games, video on demand (VoD), TV over Internet, e-learning, software updates, database replication and broadcasting stock quotes. Nevertheless, the lack of security in the multicast communication model obstructs the effective and large scale deployment of such strategic business multi-party applications. This limitation motivated a host of research works that have addressed the many issues relating to securing the multicast, such as confidentiality, authentication, watermarking and access control.

These issues must be seen in the context of the security policies that prevail within the given circumstances. For instance, in a public stock quotes broadcasting, while authentication is a fundamental requirement, confidentiality may not be. In the contrary case, both authentication and confidentiality are required in video-conference applications. In this paper, we focus on a keystone component of any secure multicast architecture over wired networks: group key management.

II. GROUP COMMUNICATION CONFIDENTIALITY

In this section, we will use a simple scenario to introduce the challenging issues relating to group confidentiality and key management. We consider a source that sends data to a set of receivers in a multicast session. The security of the session is managed by two main functional entities: a Group Controller (GC) responsible for authentication, authorization and access control, and a Key Server (KS) .

Responsible for the maintenance and distribution of the required key material. Note that these two functions can be implemented over a single physical entity or over different physical entities depending on the key management architecture. Depicts this simple scenario.

To ensure confidentiality during the multicast session, the sender (source) shares a secret symmetric key with all valid group members, called Traffic Encryption Key (TEK). To multicast a secret message, the source encrypts the message with the TEK using a symmetric encryption algorithm.

Upon receiving the encrypted multicast message $\{m\}_{TEK}$, each valid member that knows the TEK can decrypt it with TEK and recover the original one. To avoid that a leaving or an ejected member from the group, continues to decrypt

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

the secret multicast messages, the KS must generate a new TEK and securely distribute it to all remaining group members except the leaving one.

This operation is called re-keying. The KS shares a secret key called Key Encryption Key (KEK_i) with each member m_i (cf. To re-key the group following a leave from the group, the KS generates a new TEK: TEK , and sends it to each member m_i (except the leaving one) encrypted with its corresponding KEK_i. Thereby, the leaving member cannot know the new TEK and hence will not be able to decrypt future multicast messages of this session.

Hence all old members can recover the new TEK: TEK . Then, the KS encrypts TEK with the secret KEK_j that it shares with the new member m_j and sends it to him to recover TEK which is required to decrypt the multicast messages.

The maintenance and the distribution of the keys involved in re-keying and encryption is commonly called : Group Key Management. In this illustrative protocol, re keying induces a O(n) re-key message after each leave from the group, where n is the number of group members. It also induces a storage of O(n) keys (1 TEK + n KEK_i) at the KS during the whole secure multicast session. Since each membership change in the group requires re-keying and the group may be highly dynamic, one of the challenges of group key management is how to assure re-keying using the minimum bandwidth overhead without increasing the storage overhead. Proposed solutions in the literature, as we will see in the following sections, trade bandwidth overhead for storage overhead to achieve the best overall performance.

III. GROUP KEY MANAGEMENT REQUIREMENTS

Efficient group key management protocols should take into consideration miscellaneous requirements. Figure 2 summarizes these requirements from four points of view: security, quality of service, KS's resources, and group members' resources.

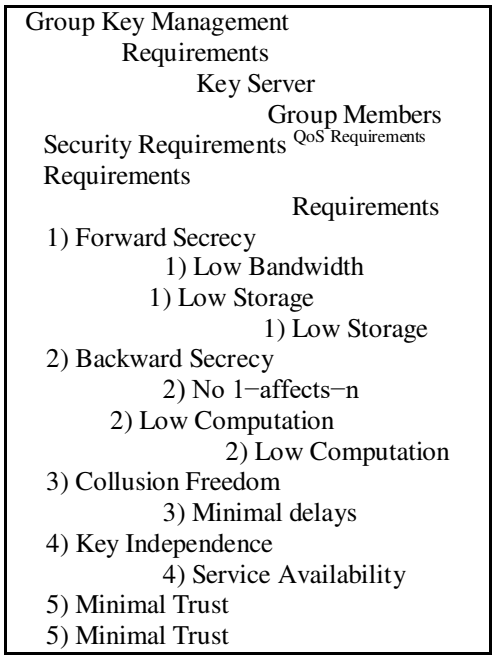


Fig. 2. Group Key Management Requirements

Security requirements:

1. Forward secrecy requires that users who left the group should not have access to any future key. This ensures that a member cannot decrypt data after it leaves the group. To assure forward secrecy, a re-key of the group with a new TEK after each leave from the group is the ultimate solution.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

2. Backward secrecy requires that a new user that joins the session should not have access to any old key. This ensures that a member cannot decrypt data sent before it joins the group. To assure backward secrecy, a re-key of the group with a new TEK after each join to the group is the ultimate solution.
3. Collusion freedom requires that any set of fraudulent users should not be able to deduce the current traffic encryption key.
4. Key independence: a protocol is said key independent if a disclosure of a key does not compromise other keys.
5. Minimal trust: the key management scheme should not place trust in a high number of entities. Otherwise, the effective deployment of the scheme would not be easy.

Quality of service requirement:

1. Low bandwidth overhead: the re-key of the group should not induce a high number of messages, especially for dynamic groups. Ideally, this should be independent from the group size.
2. 1-affects-n: a protocol suffers from the 1-affects-n phenomenon if a single membership change in the group affects all the other group members. This happens typically when a single membership change requires that all group members commit to a new TEK.
3. Minimal delays: many applications that are built over the multicast service (typically, multimedia applications) are sensitive to jitters and delays in packet delivery. Therefore, any key management scheme should take this into consideration and hence minimizes the impact of key management on the delays of packet delivery.
4. Service availability: the failure of a single entity in the key management architecture must not prevent the operation of the whole multicast session. Other requirements:

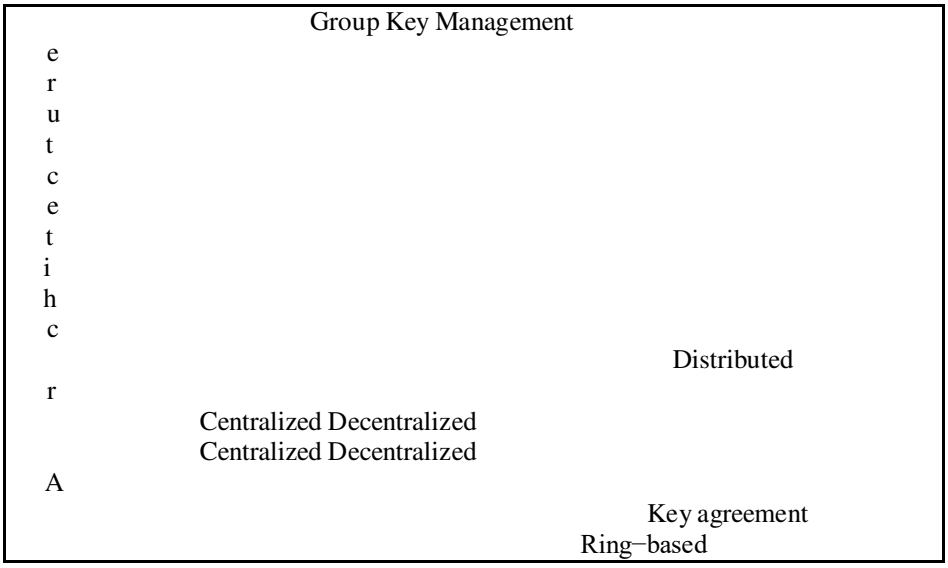
1. The key management scheme must not induce neither high storage of keys nor high computation overhead at the key server or group members.

IV. GROUP KEY MANAGEMENT

A critical problem with any re-key technique is scalability: as a rekeying process should be triggered after each membership change, the number of TEK update messages may be important in case of frequent join and leave operations. Thereby, some solutions propose to organize the secure group into subgroups with independent local TEKs. This reduces the impact of re-keying, but requires data transformation at the borders of subgroups as we will see in the following sections. Therefore, we can classify existing solutions into two approaches: the Common TEK approach and the TEK per sub-group approach as illustrated in figure 3. In what follows, we present each class of protocols and we further refine the classification in order to highlight the underlying common concepts and mechanisms. We will illustrate each identified sub-category with relevant protocols from the literature.

V. COMMON GK APPROACH

In this approach, all group members share a common Traffic Encryption Key (TEK). The management of this single key can be further classified into three classes: centralized, decentralized or distributed. Figure 3 illustrates this classification.



| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

Hierarchical
Broadcast
Hierarchical Keys Hierarchy Secrets
Broadcast
driven re–keying Time– driven re–keying
Membership–
Broadcast
Broadcast
Pairwise keys
K
Cooperation
Cooperation
Cooperation
Cooperation
Cooperation
Cooperation
Secrets
E
T
Ingemarson et al.
Fiat et al.
Octopus
GKMP Secure Lock LKH
SMKD
MARKS
n
Burmester et al.
STR
GDH
IGMP
OFT
Kronos
o
Poovendran et al.
CKA
DH–LKH
Hydra DEP
m
Canetti et al.
Dunigan and Cao
m
D–LKH
Baal
ELK
Chu et al.
o
D–OFT
C
CFKM
D–CFKM
Wong et al.
Keystone
Iolus Yang et al.
K
p
E

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

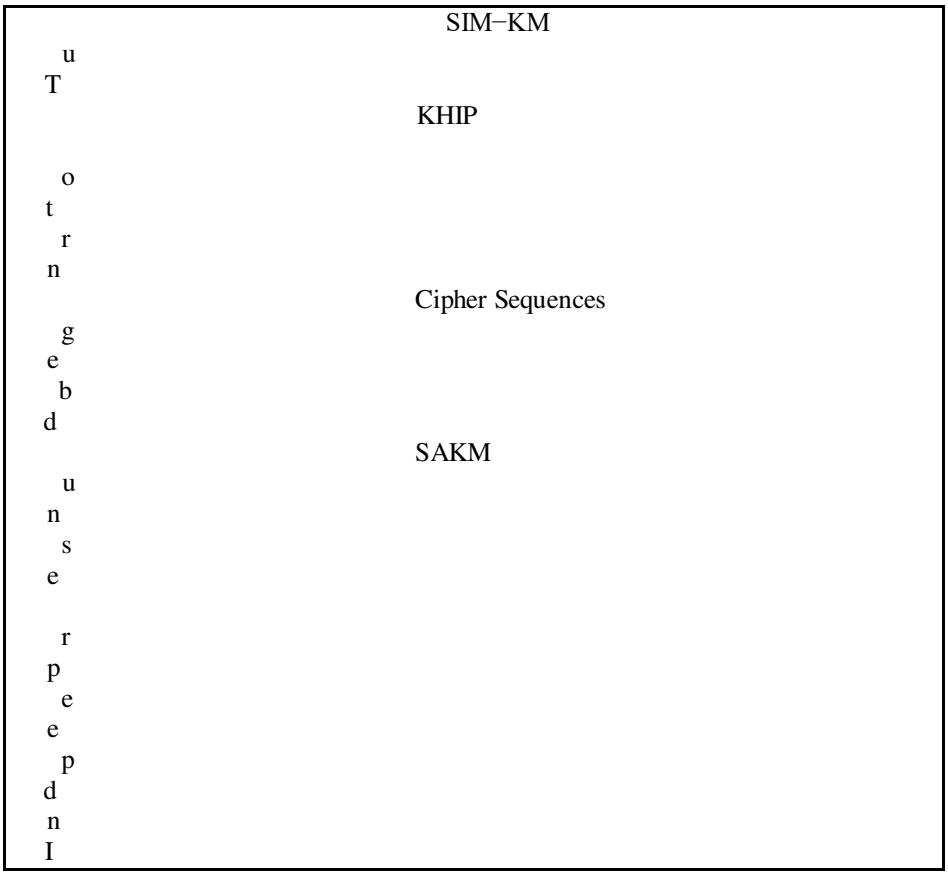


Fig. 3. Taxonomy of Common TEK Group Key Management Protocols

A. Centralized Protocols

In this approach, the key distribution function is assured by a single entity which is responsible for generating and distributing the traffic encryption key (TEK) whenever required. In figure 3, centralized protocols are further classified into three sub-categories depending on the technique used to distribute the TEKs. In what follows, we present each sub-category:

A.1 Pairwise Keys

In this sub-category of protocols, the Key Server shares a secret key with each group member. These keys are generally called: Key Encryption Keys (KEK) and are used to establish secure channels between the KS and each member in order to re-distribute the current TEK securely when ever required.

Group Key Management Protocol (GKMP): Harney and Muckenhirn [25, 26] proposed the Group Key Management Protocol (GKMP) that uses this approach. The key server shares a secret key with each valid group member (KEKs). In GKMP, the key server generates a Group Key Packet (GKP) that contains two keys: a Group TEK (GTEK) and a Group KEK (GKEK). The GTEK is used to encrypt the traffic and the GKEK is used to secure the distribution of a new GKP whenever required. When a new member joins the session, the key server generates a new GKP (which contains a new GTEK to assure backward secrecy) and sends it securely to the new member en crypted with the KEK established with this new member, and sends it to the other members encrypted with the old GTEK. The key server refreshes the GKP periodically and uses the GKEK for its distribution to the group members. When a member leaves the group, the key server generates a new GKP and sends it to each remaining member encrypted with the KEK that it shares with each member. Thus to assure forward secrecy, GKMP requires O(n) re-key messages for each leave from the group. Therefore, this solution does not scale to large groups with highly dynamic members.

Dunigan and Cao [21] proposed a similar protocol that suffers from the same issues. Poovendram et al. [38] have also proposed a similar scheme to GKMP, where authentication and authorization functions are delegated to other group members rather than centralized at the same group controller entity.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

Hao-hua Chu et al. protocol: In the solution proposed by Hao-hua Chu et al. in [14], a Group Leader shares a secret Key Encryption Key (KEK) with each group member. To send a secret multicast message m, the sender encrypts m with a random key k. Then, the sender encrypts k with the secret KEK that it shares with the group leader, and sends it to the group along with the encrypted message. Upon receiving the message, receivers cannot decrypt it since they do not know the random key k. When the leader receives the message, it decrypts k using the key that it shares with the source and constructs a validation message which contains k encrypted with each KEK that the leader shares with a valid group member (excluding the departing members). Upon receiving the validation message, each receiver decrypts k using its KEK and hence decrypts m which was encrypted using k. This protocol has the drawback to require the transmission of the validation multicast message by the group leader, with a size in the order of O(n) (n being the number of current valid group members), after each time the source sends a message to the group.

A.2 Broadcast Secrets Approach

In this sub-category of protocols, the re-keying of the group is based on broadcast messages instead of peer to peer secret transmissions.

Secure Locks: Chiou and Chen [13] proposed Secure Lock; a key management protocol where the key server requires only a single broadcast to establish the group key or to re-key the entire group in case of a leave. The protocol relies on the following theorem:

Theorem 1: Chinese Remainder Theorem Let $m_1,...,m_n$ be pairwise relatively prime positive integers, and let $a_1,...,a_n$ be any integers. Then the system of linear congruences in one variable given by:

$$x \equiv a_1 \bmod m_1$$
$$...$$
$$x \equiv a_n \bmod m_n$$

has a unique solution modulo $M = m_1 \times m_2 \times ...m_n$. The unique solution is: $x = \sum_{i=1}^{n} a_i M_i y_i \bmod M$, where $M_i = M/m_i$ and $y_i = M^{-1}_i \bmod m_i$.

In this protocol, the key server assigns a positive integer m_i to each member and shares a secret value k_i with each of them. When the server wants to send a message to the group, it generates a random value K and uses it to encrypt the message. Then, it encrypts K with each secret k_i and obtains the set $\{K_i\}$ of the encryptions of K ($K_i = \{K\}_{ki}$). Then the server computes a lock M which is the solution to the equation system:

$$M \equiv K_1 \bmod m_1$$
$$...$$
$$M \equiv K_n \bmod m_n$$

Then the server multicasts the lock M as well as the encrypted message with K. Upon reception of the lock M, each member recovers the encryption key $K = \{M \bmod m_i\}_{ki}$, and hence decrypts the received message. Only members whose secret k_i and its corresponding positive integer m_i are included in the computation of the lock M, can recover the decryption key K.

This protocol minimizes the number of re-key messages. However, it increases the computation at the server due to the Chinese Remainder calculations before sending each message to the group.

A.3 Hierarchy of Keys Approach

We showed that in the pairwise keys approach, re-keying induces a high number of update messages (in the order of O(n), with n being the number of group members). This is due mainly to the fact that the key server establishes a secure channel individually with each member and uses this channel to distribute the TEK updates. In order to reduce the number of update messages, in this sub-category of protocols, the key server shares secret keys with sub groups of the entire secure group in addition to the individual channels. Then, when a member leaves the secure session, the key server uses the secret sub-group keys, that are unknown by the leaving member, to distribute the new TEK. Thereby, sub-group secret keys allow to reduce the required number of update messages. In what follows we present some protocols that use this concept for re-keying:

Local Key Hierarchy (LKH): Independently, Wong et al. [49, 50] and Wallner et al. [48] proposed the Logical Key Hierarchy (LKH) protocol. In LKH, the key server maintains a tree of keys. The nodes of the tree correspond to KEKs and the leaves of the tree correspond to secret keys shared with the members of the group. Each member holds a copy of its leaf secret key and all the KEKs corresponding to the nodes in the path from its leaf to the root. The key corresponding to the root of the tree is the TEK. For a balanced binary tree, each member stores at most $1+\log_2(n)$ keys, where n is the number of group members.

This key hierarchy allows to reduce the number of re-key messages to O(logn) instead of O(n) in GKMP. Example: Let us consider a multicast group with six members $\{u_1, u_2, u_3, u_4, u_5, u_6\}$. The key server builds a hierarchy of keys as shown in figure 4. Each member owns a secret key which is a leaf in the tree as well as all the keys on its path to

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

the root. The root represents the TEK shared by the members. The other keys are used to reduce the required rekeying messages.

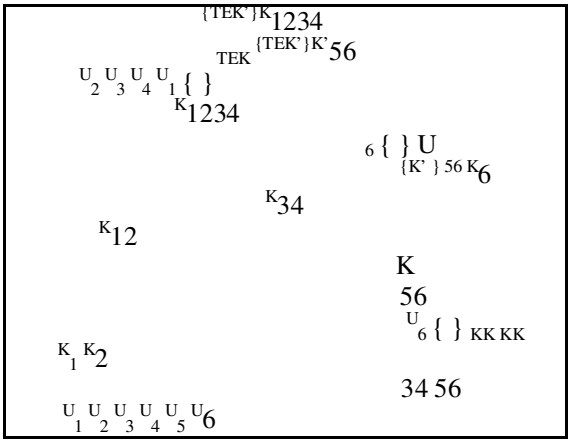


Fig. 4. key Hierarchy

Let us assume that u_5 leaves the group, KS updates k_{56} into k_{56}, sends k_{56} to u_6 encrypted with k_6. TEK is updated into TEK' and sent to $\{u_1,u_2,u_3,u_4\}$ encrypted with k_{1234} and to u_6 encrypted with k_{56} and hence only three messages are required instead of five messages if

where lef t(i) and right(i) denote respectively the left and right children of node i, and g is a one-way function. The result of applying g to a key k: g(k), is called the blinded key version of k.

In this protocol, each member maintains its leaf secret key and its blinded sibling key and the set of blinded sibling KEKs of its ancestors. Figure 5 illustrates the ancestors and their corresponding sibling keys of member u2.

VI. CONCLUSION

Based on this finding, it can be concluded that the content classification performance will be improved with enhancements as a feature selection. The second finding is that the use of the interleaved hybridization generated better optimal features for the classifier than using all the features From this observation, it can be stated that content classification can be better performed using all the optimal features generated by the interleaved hybridization.

FUTURE ENHANCEMENTS

Our future work targets the botnet phenomena in securing cloud data by Group Key Management protocol

REFERENCES

[1] K. Almeroth and M. Ammar. Collecting and modelling the join/leave behaviour of multicast group members in the Mbone. Symposium on High Performance Distributed Computing, 1996.

[2] D. Balenson, D. McGrew, and A. Sherman. Key Management for Large Dynamic Groups : One-Way Function Trees and Amortized Initialization. draft-balenson-group keymgmt oft-00.txt, February 1999. Internet-Draft.

[3] A. Ballardie. Scalable Multicast Key Distribution, May 1996. RFC 1949.

[4] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing protocol specification, September 1997. RFC 2189. [5] T. Ballardie, I.P. Francis, and J. Crowcroft. Core Based Trees: an Architecture for Scalable Inter-domain Multicast Routing. ACM SIGCOMM, pages 85–95, 1993.

[5] B. DeCleene, L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, and C. Zhang. Secure group communications for wireless networks. MILCOM, June 2001.

[6] C. Becker and U. Wille. Communication complexity of group key distribution. 5th ACM Conference on Computer Communications Security, November 1998.

[7] C. Boyd. On key agreement and conference key agreement. Information Security and Privacy: Australasian Conference, LNCS(1270):294–302, 1997.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904101 |

[8] B. Briscoe. MARKS: Multicast key management using arbitrarily revealed key sequences. 1st International Workshop on Networked Group Communication, November 1999.

[9] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. EUROCRYP'94, LNCS(950):275–286, 1994.

[10] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast Security: A taxonomy and Efficient Constructions. IEEE INFOCOM, pages 708–716, March 1999.

[11] G. Chaddoud, I. Chrisment, and A. Shaff. Dynamic Group Communication Security. 6th IEEE Symposium on computers and communication, 2001.

[12] Y. Challal, H. Bettahar, and A. Bouabdallah. SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications. ACM SIGCOMM Compute

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

[14] 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.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details