ADVANCES in NATURAL and APPLIED SCIENCES

ISSN: 1995-0772
EISSN: 1998-1090
2016 September 10(13): pages 9-21

Published BY **AENSI Publication**
http://www.aensiweb.com/ANAS
**Open Access Journal**

# Cost effective multicast-unicast based key management for secure group communication

**[1]Vinoth Chakkaravarthy G, [1]Dr. Alli P, [2]Raja Lavanya**

*[1]Department of Computer Science and Engineering, Velammal College of Engineering and Technology, Madurai 625009, India.*
*[2]Department of Information Technology, Thiagarajar College of Engineering, Madurai 625015, India.*

Received 7 June 2016; Accepted 12 September 2016; Available 20 September 2016

**Address For Correspondence:**
Vinoth Chakkaravarthy G., Velammal College of Engineering and Technology, Department of Computer Science and Engineering, 625009, Madurai, India.
Mobile No: 9994448831; E-mail : vc_04@yahoo.co.in

**Copyright © 2016 by authors and American-Eurasian Network for Scientific Information (AENSI Publication).**
**This work is licensed under the Creative Commons Attribution International License (CC BY).**
http://creativecommons.org/licenses/by/4.0/

Open Access

## ABSTRACT

The rapid increase in the Internet usage in various applications like healthcare, military, home, online collaborative chat and video conferencing, etc. increases the popularity and attention of the group communication. For a group communication, multicasting is a more suitable and efficient mechanism rather than all types transmission methods because, the bandwidth requirement of the multicast is less when compared to all. From the security point of view, multicast needs lot of improvement to provide secure communication. To ensure the security of the group communication without affecting the performance, key management is more suitable and efficient framework. Security and efficiency are the two contradictory objectives to be achieved in designing a key management framework for a resource constraint wireless group communication. In this paper, we propose a multicast – unicast based key management framework to improve the performance in terms of communication, storage and computation costs without affecting the security. The proposed key management is a decentralized framework which works in three dimensions. The first dimension deals with the scalability problem, which is addressed by dividing the network into subgroups and each subgroup is managed by a subgroup manager. The subgroup manager securely shares the secret keys with each member in a unicast manner. The SM securely multicast the encrypted TEK to each member. The second dimension aims at reducing communication, computation and storage costs of the proposed technique during join and leave operation. The third dimension focuses on the security aspects of the proposed scheme. To fulfil the objectives, the proposed technique has been implemented using a tree based key management scheme and the performance is analysed in terms of communication, storage and computation costs with the existing schemes SKDC, OFT and SKD. The performance analysis is carried out with $2^5$ members for every scheme. The communication cost of proposed MUKD is very low since only one TEK is transmitted during membership changes. Regarding computation cost, MUKD needs one key encryption and one key decryption. Hence the proposed schemes experiences very low computation cost. The number of keys stored by subgroup manager and members is also considerably reduced in MUKD. The results confirm that the proposed MUKD acquires the good performance in terms of communication, computation and storage costs without affecting the security.

**KEYWORDS:** Group Key, Management Key, Refreshment, Overhead reduction

## INTRODUCTION

Due to rapid increase in the Internet usage in various applications like healthcare, military, home etc. multicast for group communication is getting more popular. Multicast communication reduces the required bandwidth. So it is an efficient way of transmitting the data simultaneously to multiple recipients. Hence multicast is more suitable and efficient for group communication which saves network bandwidth [1,2]. But multicast communication has lot of security constraint. From the security point of view, multicast needs lot of

**To Cite This Article:** Vinoth Chakkaravarthy G, Dr. Alli P, Raja Lavanya., Cost effective multicast-unicast based key management for secure group communication. **Advances in Natural and Applied Sciences**. 10(13); Pages: 9-21

improvement to provide secure communication. Hence, for a multicast group communication, a demand to improve the security features is constantly increased [18]. To satisfy the mandatory security features like confidentiality, authentication and integrity the group communication needs more encryption / decryption cycles. This liberal use of cryptographic functions affects the performance the wireless network environment. Most of the existing methods practice with strong cryptographic function that uses more encryption/decryption cycles to improve the security. But security and efficiency are the two contradictory objectives to achieve [21]. Involving more encryption/decryption increases the communication, computation and storage costs. Communication cost refers to a number of message exchanges and keys shared for key distribution. Storage cost refers to number of keys stored at server and members. Computation cost refers to the number of encryption and decryption needed to distribute the group key. To satisfy the mandatory security features like confidentiality, authentication and integrity without affecting the efficiency, an appropriate key management scheme for group communication is required [11].

Group Key Management [GKM] is a primary security service for a multicast oriented group communication. GKM mainly focuses on group key generation, distribution and key refreshment during rekeying process. For managing the group, whenever group membership change occurs, a new group key has to be generated and securely delivered to the members to ensure forward and backward secrecy. Other than group membership changes, rekeying can be done for the following reasons: Periodical Rekeying, expiration of keys, Compromised keys. Many researchers have proposed different group key agreement protocols for generating group key. Jian Zhou proposed a key management scheme which is based on one-encryption key and multi-decryption key protocol [6]. In this scheme, each user has unique decryption key that corresponds to a common group key which is used for encryption. Young-Hoon Park proposed a key management protocol for secure multicast [25,23]. Here, a common group key can be used for both encryption and decryption. Using the group key, the message for that group is encrypted and transmitted once. The members having the group key can decrypt the multicast message. A well-known mechanism to generate the common group key is Diffie-Hellman protocol which depends on exponential computation [8,3]. The following are the factors that affect the performance of the group key communication: 1-affects-n phenomenon, Scalability, Operational efficiency [4].

To achieve maximum security and efficiency, the group key should be securely and handily distributed among the group members, so that the group members can collaboratively work with each other without bothering about the leakage of information [4,7,23]. P. Vijayakumar, proposed an extended Euclidean algorithm for a secure key distribution for centralized secure multicast communication [14]. Authors of paper [19,20] proposes encrypted polynomial based secure group key distribution techniques for inter-group and intra- group communication. Yanji Piao proposed a polynomial based key distribution without encryption, which is efficient for intra group communication [24]. We confirm that we have used the version of this polynomial based key distribution without encryption in the proposed system.

For managing the group, whenever group membership change occurs, a new group key has to be generated and securely delivered to the members to ensure forward and backward secrecy. In a highly dynamic environment, frequent rekeying causes too many encryption and decryption in key distribution [5]. Most of the existing key management methods are based on key trees where the server is responsible for managing the group key. This result high computation, communication and storage costs and causes burden to the network. Many researchers have proposed different tree based dynamic group key management approaches for multicast namely: LKH, OFT and SKD. To address the scalability problem in the tree based key management scheme, Abbas Mehdizadeh proposed multicast – unicast key management method for wireless IPv6 network, where the network is divided into subgroups and each subgroup is managed by a subgroup manager(SM) [2,15]. This significantly reduces the communication and computation cost of rekeying process. The group communication using GKM may subject to the following vulnerabilities:

- Eavesdropping of group message as well as cryptographic keys.
- Modification of the original message by changing a part of the message or inserting the new message.
- Deleting the group message in order to avoid valid member to receive.
- Masquerading as a valid member.
- The inside attack is an attack from the legitimate member to impersonate the secret communication between individual member and the manager
- The outside attack is an attack from non-member or previous member to eavesdrop the group information.

In this paper, we propose a decentralized key management for reducing the overhead without affecting the security in the wireless group communication networks.

*1.1 Objectives and contributions of the work:*

The main objective of the work is to develop a multicast – unicast based key management framework to achieve efficiency in terms of communication, computation and storage costs without compromising the security. To prevent the network from being attacked (both inside and outside), the proposed work achieves the

following the security requirements namely: Group Secrecy (ensures the communication inside the group is only by the group members in a confidential manner), Group Authentication (allows only the legitimate members of the group to be involved in the communication), Integrity (no alternation during transmission inside the group) and key freshness (the subgroup is refreshed by receiving the new group key during join and leave). The following security metrics can be used to ensure the security aspects of the proposed work: forward secrecy, backward secrecy, group key secrecy and key independence.

*Contributions of the work:*

This paper makes the following technical contributions to the group key management in the wireless environment:

- It performs the critical analysis of different group key management schemes in terms of communication, computation and storage costs.
- It proposes decentralized key management frameworks to address the following things:
- Scalability
- 1 affect n problem
- It uses the both unicast and multicast transmission for the key management.
- The communication, computation and storage costs of the proposed schemes are calculated and compared with the existing schemes.

*1.2 Limitations of the work:*

- Cost for electing subgroup manager and group key generation is not considered for efficiency.
- The efficiency parameters other than communication, computation and storage costs are considered for performance analysis.

*1.3 Structure of the paper:*

The remaining part of this paper is organized as follows: Section 2 gives brief description about the existing key management schemes: OFT SKDC, Iolus and SKD. Section 3 presents the work related with a proposed problem statement which covers formation of sub group, group key generation and group key distribution. Section 4 provides information about rekeying process due to join and leave operation. Section 5 describes the comparative analysis of MUKD with the existing schemes. Section 6 is the conclusion of the research work.

*2. Related works:*

According to topological structure, the group key management schemes can be classified into three categories namely: centralized, decentralized and distributed key management scheme. In centralized key management scheme, a central server is responsible for managing the whole group. The server is still in a security bottleneck and the rekeying cost is higher because, more intermediate keys are rekeyed. In decentralized key management scheme, the members of a multicast clique are fragmented into many smaller subgroups that are governed by variety of subgroup controllers. This approach eliminates the single point vulnerability problem as well as one –affect – N issue and significantly reduces the rekeying cost. In distributed key management scheme, each member must contribute in the generation of group key by sharing the part of its key material. In this scheme, rekeying cost is high since all the members again contribute to generation of the new group key. Three existing key management schemes namely: OFT (centralized), Iolus, MUKD (decentralized), SKD (distributed) are considered.

*2.1 Existing key management Schemes:*

*2.1.1 One – way Function Tree:*

A. Sherman proposed that OFT uses balanced binary tree for implementing the key management where bottom-up approach is used for key generation (Sherman, A ,2003), (Vinoth Chakkaravarthy, G ,2015)). In OFT, key tree is solely created and managed by the root. Because of its bottom up approach, the complexity depends on the height (h) of the tree. For a d-ary tree, number of keys transmitted corresponds to the height during leave and join. OFT performs more encryption and decryption to achieve maximum security. The server performs h * 2 encryptions for join and h encryptions for leave. The members perform h decryptions for both join and leave. The server stores 2n-1 keys and h keys are stored by members.

Demerits of OFT:

- The broadcasting nature of the changed information leads to high communication cost
- Many intermediate keys are rekeyed which results in high computation cost.

*2.1.2 Simple Key Distribution Center (SKDC):*

Wong et.al., 2002 proposed SKDC, where the server is responsible for the process of key generation, key distribution and key refreshment of a group. This framework uses the key tree which is shown in figure, which addresses the scalability problem for large groups where the arrival and departure happens frequently Whenever

a new node joins, a new group key is generated and the same is sent to newly joined member through unicast or multicast. Here, the server performs random key generation and key encryptions. At the same time, members (old and new) perform one key decryption. When a member leaves from the group, all the keys that the departed node had access to must be changed. When a member leaves from the group, the server performs key generation and (n-1) key encryption, where n is the number of members. In this approach, the server stores (n+1) keys for a group of n members. According to the Tan Zhu. 2013 statement, the computation, communication and storage costs of SKDC are proportional to the number of members.

*2.1.3 Iolus:*

Mittra S. Iolus proposed that, in Iolus, a large group is divided into many smaller subgroups and each subgroup is managed by a sub group controller [10,17]. Two objects namely: Group Security Controller for top-level subgroup and Group Security Agent for controlling each subgroup are involved. For a data transmission among groups, decryption and re-encryption is performed.

Demerits of Iolus:

- Many sub groups lead to lot of decryption and re-encryption.
- Data can't be decoded, if the connected subgroups fail to refresh each other with the new key.

*2.1.4 Shared Key Derivation:*

J.C Lin proposed that Shared key derivation [23] is a combination of centralized and distributed key management [9,17]. Initially the server distributes the group key to the members. Later when a member joins or leaves the group, the new group key is generated by members by using old key. For a d-ary tree, h keys are transmitted during join. During leave, key updating is needed only for the children of the intermediate nodes. Therefore (d-1) * h keys are transmitted. The server performs h encryptions for join and (d-1) * h encryptions for leave. The members exactly perform the same number of decryptions for join and leave. Regarding the storage of keys, members store h keys but the server depends on number of users and d.

Demerits of SKD:

- The communication cost is higher because the cost is equal to the height (h) of the tree.
- The computation cost is higher because during rekeying, the keys of all the nodes that lie on the path between the member and server needs to be updated.

*3. System Model:*

In MUKD, it is assumed that the network can be logically viewed as a key tree which is shown in Fig. 1. The members (N) are in the leaf nodes of the key tree, the SMs are in the immediate higher level of the members and heads of the SMs (HSM) are in the next higher level. The maximum hop count between the members and the SM in the key tree is one.

*3.1 Conceptual framework:*

To handle the scalability issues of the dynamic environment, a network is fragmented into subgroups (S) and each subgroup is managed by a SM. Each member generates the hash code using its MAC code and shares the hash code to the subgroup managers. The SM forwards the hash code to the root and can be used for authentication during joining and leaving of a member. The subgroup manager generates its own key and the secret keys for each member and securely distributes to the respective members using unicast. Then the subgroup manager multicast its key to the members. The HSM assist the inter-group key distribution for the interaction between the subgroups. After the selection of the manager for all subgroups, next election is for choosing the head of different subgroup managers in the next higher level. The same process is continued till the root is found. For a subgroup, the subgroup manager alone is responsible for key management process. The heads of the subgroup manager are not involved in any key management process and are involved in the key management process to assist the intergroup communication. The root can be used only for authenticating the members to prevent inside and outside attacks.

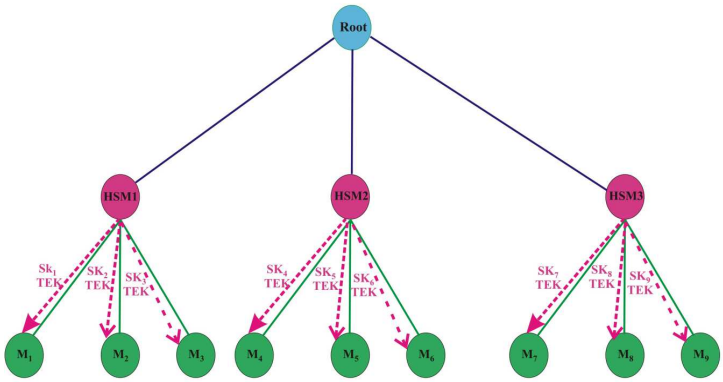


**Fig. 1:** System model

*3.2 Process flow diagram:*

The overall process of MUKD is represented in Fig. 2

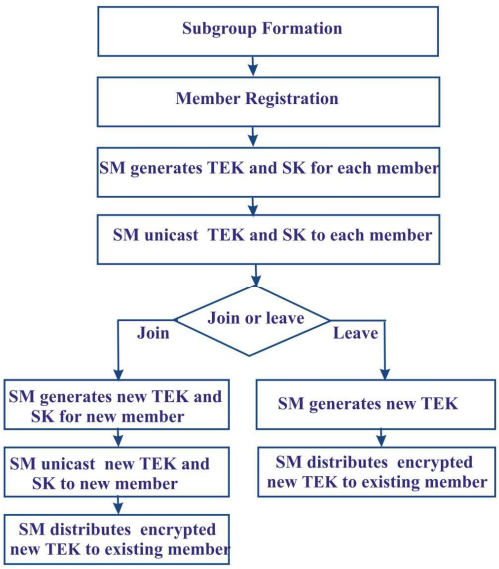


**Fig. 2:** Process flow diagram

*3.3 Formation of subgroup:*

The formation of subgroup is initiated by selecting the subgroup manager.

Procedure for SM selection:

- In the leaf node level, all the members having the hop count between them is one can form the subgroup.
- These nodes send a hello message which contains the MAC address to its neighboring node. If the members are the computer nodes, then each node shares its unique 6byte MAC address through hello messages.
- Each node compares its MAC value with the received one and the priority is given to the largest weighted node.
- Finally the largest weighted node is selected as the manager of that subgroup.

After the subgroup manager selection for all subgroups is completed, the next process is to select the Heads of the Subgroup Manager.

Procedure for HSM selection:

- Each subgroup manager shares its MAC to another manager using a multicast frame containing the hello message.
- The same procedure which is used for the election of subgroup manager is applied and finally the largest weighted node is elected as HSM.
- Immediately the new manager is elected for the subgroup where the previous SM is now elected as a HSM.

The same procedure is continued to find the root. After the logical tree construction, the root generates a data base in the form of hash table as shown in Table for authenticating the members during join and leave.

The hash value is generated based on the following procedure:

- A hash function H takes a variable size message M and produces a fixed length hash value hm ( H(M) = $h_m$)

The hash table contains the hash value of the existing members in the subgroup. Based on these hash values, the root takes the decision about accepting or denying the joining request.

*4. Key generation:*

The key generation process is the responsibility of the subgroup manager. The SM generates two types of keys namely:

- Secret keys for each member (Individual Key)
- Ticket Encryption Key (Subgroup manager key)

The Secret Keys are also called as Individual Keys and are used for individual secure communication between the member and the SM. The TEK is a common key for the group and is used for the group communication. MUKD applies Pseudorandom Number Generators (PRNGs) for the generation of TEK and IKs of the members. The linear congruential method is used for the generation of the random numbers (Lehmer 1951). The algorithm is parameterized with four numbers, as follows:

| | | | | |
|---|---|---|---|---|
| m | - | the modulus | where | $m > 0$ |
| a | - | the multiplier | where | $0 < a < m$ |
| c | - | the increment | where | $0 \leq c < m$ |
| X0 | - | the starting value, or seed | where | $0 \leq X_0 < m$ |

The sequence of random numbers {Xn} is obtained through the following iterative equation:

$Xn+1 = (aXn + c) \bmod m$

*5. Key distribution:*

After the key generation, the generated keys must be securely distributed to the members. Initially, SM delivers the secret keys and TEK individually to each member using unicast addressing approach. During membership changes, the SM updates its TEK and securely distributes to each member using encryption and decryption through unicast approach. The SM encrypts the TEK using each member's IK and delivers it individually to the respective members.

Procedure key distribution (members Ni, Subgroup Manager SM, Secret Keys IKi, Subgroup Manager Key TEK)

{
SM generates IKi and TEK
SM unicast IKi and TEK to Ni
Ni receives IKi, TEK and stores
}

*6. Key refreshment:*

The group membership may well change in the dynamic networking environment. In MUKD, because of subgroup arrangement, the entire group is not involved in rekeying process. Instead, the respective subgroup alone is involved in the rekeying process. The key tree represented in Fig.3 describes the rekeying process of the proposed MUOEMD as follows: When the new member M13 wants to join in a subgroup, it first sends a join request to SM4. SM4 receives the request and forwards it to the root Kr through HSMs. Kr calculates the hash value H(MAC13) and looks up at its table to grant permission or deny the request. For example, if the hash code finds matches in the hash table, then the request is accepted. If the hash code does not find any match, the request is not accepted and the respective member is blocked to join in the group. After comparison, Kr authenticates the request and this authentication is sent back to the SM4. After getting the authentication, SM4 generates new Ticket Encryption Key (TEK') and secret key (IK13). Next SM4 distributes the TEK' and IK13 to M13 using unicast approach. At the same time, SM4 encrypts the TEK' using IK10 – IK12 and sends to the members M10-M12 using unicast transmission.

Procedure join (member Ni, Subgroup Manager SMi, root Kr, Secret Keys IKi, Ticket encryption Key TEKi)

{
Ni sends join request to SMi
SMi forwards Ni request to Kr

Kr calculates hash code H(MACi)
Kr verifies for authentication
If ( Kr accepts)
{
Kr informs to SMi about the acceptance
SMi generates new TEK' and IKi
SMi unicast the IKi to new member Ni
SMi encrypts TEK' using IKi of existing Ni of that subgroup
SMi unicast the encrypted TEK' to existing Ni
}
else
rejects the request
blocks the member Ni
}

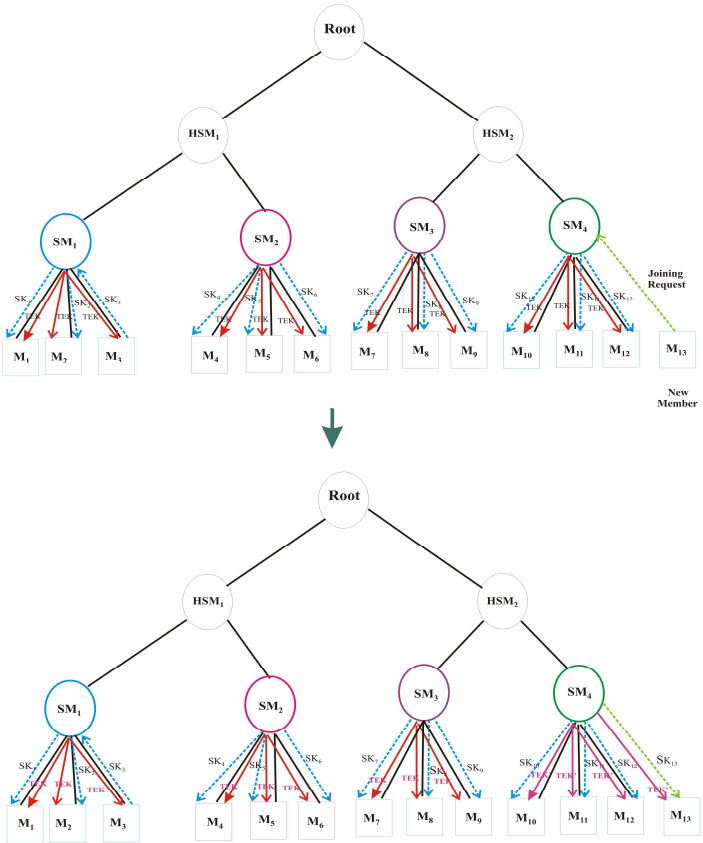


**Fig. 3:** Rekeying process

Similarly, when a member leaves, again new TEK" is generated and is encrypted using IK10 - IK12 (existing members secret keys). The encrypted TEK" is distributed to the existing members using unicast addressing approach.

Procedure join (member Ni, Subgroup Manager SMi, root Kr, Secret Keys IKi, Ticket encryption Key TEKi)
{
Ni sendsl eave request to SMi
SMi sends Ni request to Kr
Kr calculates hash code H(MACi)
Kr verifies for authentication
If (Kr accepts)
{
Kr informs to SMi about the acceptance
SMi deletes the record of Ni

SMi generates new TEK"
SMi encrypts TEK" using IKi of existing Ni of that subgroup
SMi unicast the encrypted TEK' to existing Ni
}
else
rejects the request
}

*7. Performance Analysis:*

The performance of the proposed MUKD is analyzed based on the three parameters namely: communication cost, computation cost and storage cost. The communication cost denotes the number of key transmissions required for key update. The computation cost denotes the number of encryption and decryption needed for key update during membership changes. The storage cost mentions the number key stored at subgroup manager and members. The operational efficiency of the proposed MUKD is analyzed and compared with existing schemes SKDC, OFT and SKD. Like other proposed schemes, the key management schemes are implemented using JAVA (Intel CORE i3 processor with 4 GB RAM) and the performance is analyzed with 2n users. According to the Abbas Mehdizadeh et al statement, maximum of $2^5$ (32) users are considered for every scheme.

In the logical tree structure, the parameter n represents the number of members and h represents height of the tree is used for the performance analysis. Here, h is equivalent to 1+logd n. Because of the binary tree structure, d corresponds to 2 in OFT. For SKD, 4 –ary tree (d=4) is considered. In the proposed MUKD, the communication between the subgroup is not considered for performance analysis, because many encryption and decryption is needed for key update. The performance analysis is carried out only for communication at one subgroup.

*7.1 Communication cost:*

The communication cost refers to the number of keys transmitted during join and leave operations of key management process. Table 1 and Fig. 4 illustrates the communication cost of SKDC, OFT, SKD and MUKD during join and leave operation.

Table 1 and Fig. 4 illustrates the communication cost of SKDC, OFT, SKD and MUKD during join and leave operation.

**Table 1:** communication cost of key management schemes

| Protocols | Number of message / key exchanges | | |
|---|---|---|---|
| | Join | | leave |
| | Unicast | Multicast | Multicast |
| SKDC | 1 | 1 | n-1 |
| OFT | logd(n)+1 | logd(n)+1 | logd(n)+1 |
| SKD | logd(n)+1 | logd(n)+1 | (d-1) logd(n)+1 |
| MUKD | 1 | 1 | 1 |

The communication cost between the subgroup manager and each member remains constant in MUKD. The communication cost of join and leave for MUKD is 1 because during membership changes only one TEK is updated and transmitted. From the above results, for a join operation, OFT suffers with high communication cost because of the balanced binary tree structure. When the number of user increases, the communication cost also gets increased. Hence the efficiency of OFT is worst when compared with all mentioned schemes for join operation. Both SKDC and MUKD experiences lower communication cost for both unicast and multicast in the join operation. In this regard, MUKD and SKDC have similar performance.

But during leave operation, SKDC is severely affected by the higher communication cost because SKDC transmits (n-1) keys for every change. When the number of user increases, the communication cost also drastically increases. The communication cost of SKD is also higher, because key update is needed for the children of the intermediate nodes in SKD. For this reason, when the number of user increases, the communication cost also gets increased in SKD. MUKD has lower communication cost because of single key transmission.

From these observations, in the existing schemes, all the nodes that lie on the transmitting path from member to root need to be rekeyed during key refreshment. If the number of user increases, the height of the tree also increased which leads to the increase in the communication cost of OFT and SKD. But in the proposed MUKD, only the common encryption key of the respective manager alone is rekeyed. If the number of user increases, the height of the tree never increased because the maximum hop count between the subgroup manager and the member is one. Hence, the proposed MUKD achieves superior performance compared to SKDC, OFT and SKD in terms of communication cost.

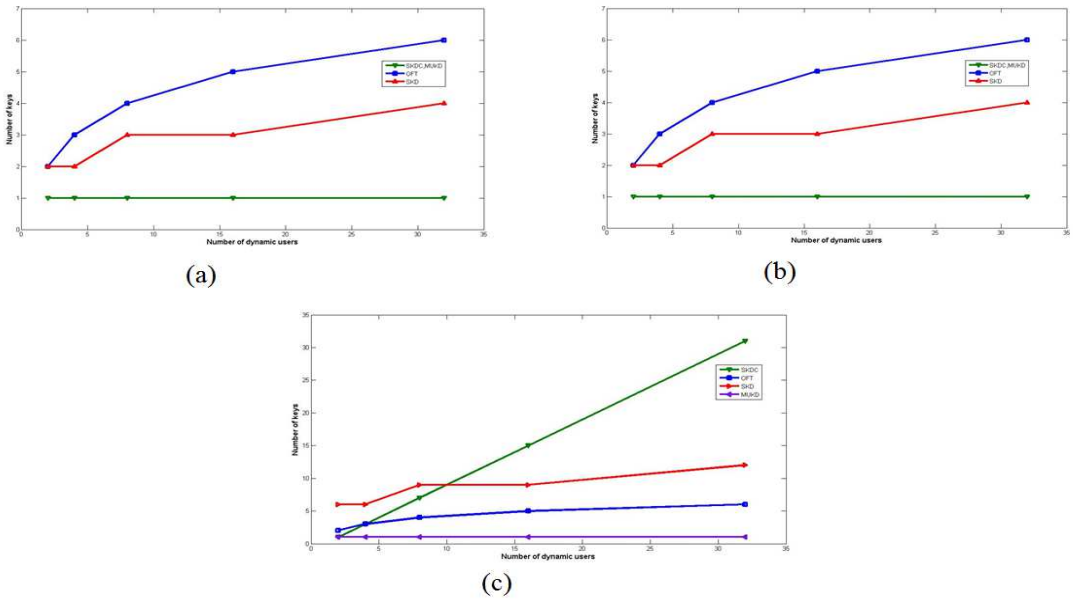


**Fig. 4:** Communication cost a) join – unicast b) join – multicast c) leave – multicast

*7.2 Storage Cost:*

The storage cost can be calculated based on the number keys stored at manager and members. Table 2 and Fig. 5 shows the storage cost of SKDC, OFT, SKD and MUKD.

**Table 2:** storage cost of key management schemes

| Protocols | Number of Message/ Keys stored | |
|---|---|---|
| | Server | Members |
| SKDC | (n+1) | 2 |
| OFT | 2n-1 | logd(n)+1 |
| SKD | (dn-1)/ (d-1) | logd(n)+1 |
| MUKD | 1 | 1 |

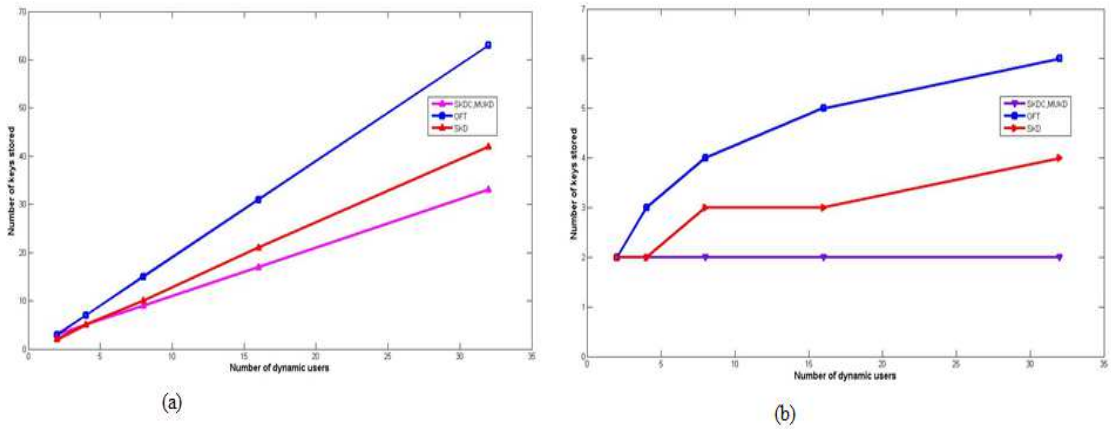


**Fig. 5:** Storage cost a) subgroup manager b) members

From the member point of view, the storage cost depends on the height of the tree. Because of the balanced binary tree structure, the storage cost of OFT is higher than other schemes. The number of keys stored by the server in OFT corresponds to the number of member-1 (n-1). In SKD, it depends on number of users (n) and the intermediate children (d). In SKDC, the number of keys stored by the server corresponds to member count along with the group key. Whenever the number of user increases, the storage cost of OFT at server gets increased which leads to the poor performance.

The storage cost of SKDC at member is similar to MUKD, but at the same time, it is higher than MUKD at server. At members, OFT suffers with high storage cost when compared with other schemes. In MUKD, the

subgroup manager stores one Ticket Encryption Key and n secret keys, whereas the members store one secret key and the Ticket Encryption Key.

Hence, the performance of MUKD is similar with SKDC and higher than OFT and SKD in terms of storage cost.

*7.3 Computation cost:*

Computation cost refers to the number of key encryptions and key decryptions needed for key distribution process. Table 3, Fig. 6 and 7 illustrate the computation cost of SKDC, OFT, SKD and PMUKD for server and members during join and leave.

For the computation cost at the subgroup manager, MUKD needs one key encryption and one key decryption. Like server, both new member and existing members need one key decryption for retrieving the key. In the existing schemes OFT and SKD the computation cost depends on the height of the tree.

At server, OFT suffers with high computation cost because of the one-way key derivation during join. For leave operation SKDC has worst computation cost since most of the encryption and decryption is handled by the server. At the same time, SKD experiences lesser computation cost than OFT during join and higher computation cost than OFT during leave. As far as members are concerned, OFT suffers with high computation cost since the number of decryption depends on the height of the tree for join and leave. SKDC has the similar performance as MUKD at the members.

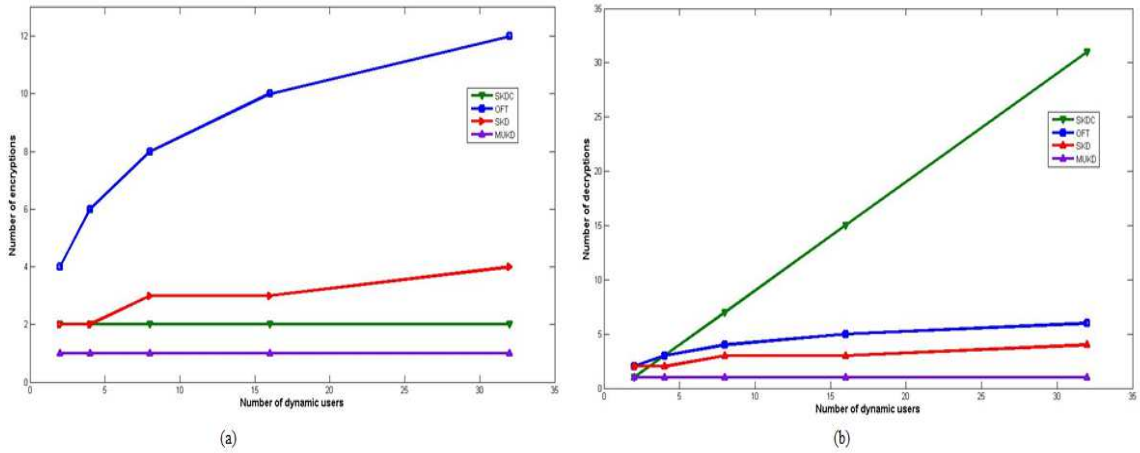


**Fig. 6:** Computation cost at subgroup manager a) join b) leave

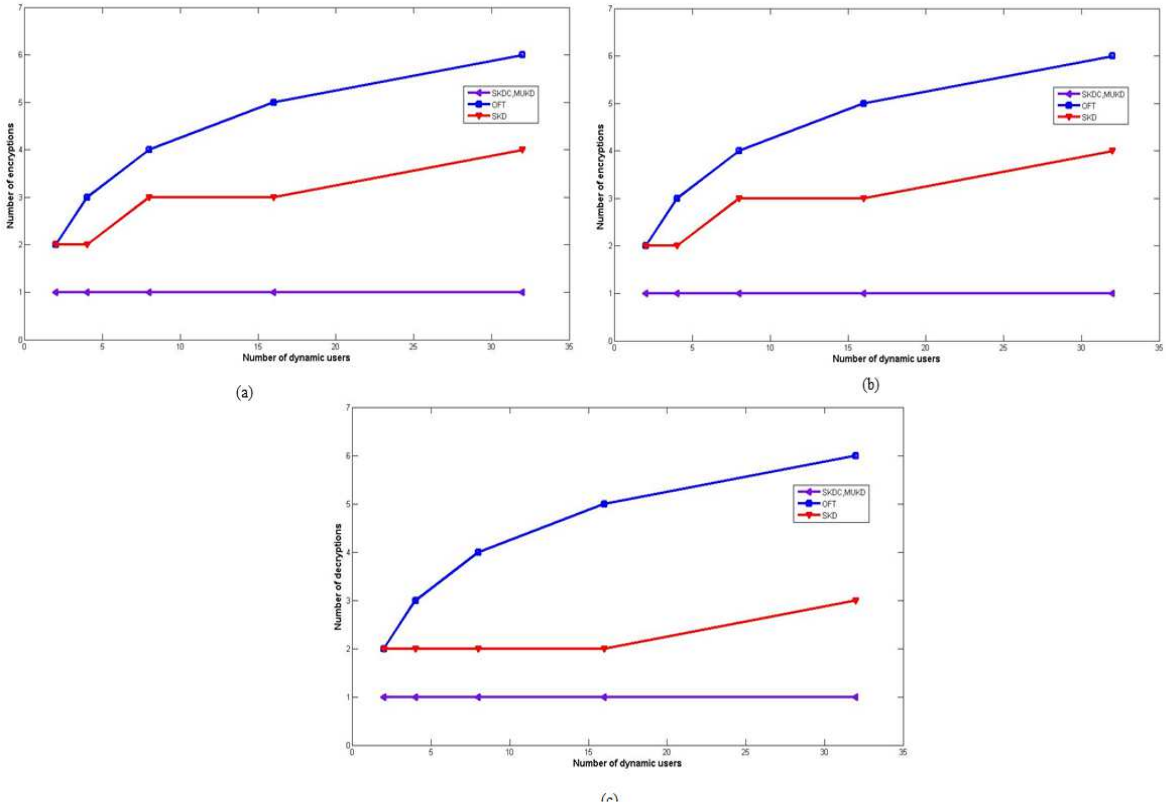

**Fig. 7:** Computation cost at members a) join- old member b) join- new member c) Leave –old member

**Table 3:** Computation cost of key management schemes

| Protocols | Number of encryption at Server | | Number of decryption at Members | | |
|---|---|---|---|---|---|
| | Join | Leave | Join | | Leave |
| | | | Old member | New member | Old Member |
| SKDC | 2 | (n-1) | 1 | 1 | 1 |
| OFT | 2( logd(n)+1) | logd(n) +1 | logd(n)+1 | logd(n)+1 | logd(n)+1 |
| SKD | logd(n)+1 | (d-1) logd(n)+1 | logd(n)+1 | logd(n)+1 | ((d-1) logd(n)+1 )/d |
| MUKD | 1 | 1 | 1 | 1 | 1 |

For both join and leave operation, MUKD offers lower computation cost compared with all the above mentioned existing schemes at the server and members. During the membership changes, the entire group need not be changed and only the subgroup alone gets updated. Thus the proposed PMUKD achieves superior performance in terms of computation cost.

*7.4 Security aspects of MUKD:*

MUKD provides the resistance against the following attacks:

Inside attack is launched by a group member who tries to intercept the individual communication of other member with SM. In MUKD, the attacker is not able access the individual communication of member with SM because the information is encrypted by the member's secret key and is unicasted to the respective member using its MAC address. The inside attacker does not receive this unicast data and is not able modify. Thus the proposed MUKD provides resistance against inside attack.

In the outside attack, the attacker may be a non-member or previous member. The non-members can do anything in the MUKD because they have to register with the Root to hear the communication. The previous member is prevented to attack with the help of forward secrecy.

Replay attack can be launched by the previous member of the group. During membership changes, each time new TEK is generated and delivered to the members. In Replay attack, the attacker tries to give the old TEK instead of new one and tries to impersonate the communication. In MUKD, when the member leaves, its record is completed removed from the SM table and the new TEK is unicasted to remaining member securely. The departed member does not receive this transmission. Hence MUKD provides résistance against replay attack.

In the group communication the keys are delivered using unicast transmission approach. The maximum hop count between the SM and the member is one and all the members are recognized as authorized members by the root and SM. Hence it is not possible for the attacker to impersonate the unicast communication to derive the key and resist the Man-In-The Middle attack

MUKD ensures the availability of service. With the help of this service, the attacker may launch the DoS attack against the SM. In MUKD, the SM fixes the threshold level for the bandwidth usage of each member. For a wireless access point having 32 members, maximum of 50 gigabits/seconds threshold level is set to each member. If any member exceeds this bandwidth very quickly, the respective member is removed from group by forcing to leave from the group. Thus MUKD resists the DoS attack.

Even if the attacker the spoofs the IP address of the member and launches the DoS attack, because of the bandwidth threshold level for each member, MUKD restricts the Spoofing attack.

*Conclusion:*

This paper proposes a decentralized key management framework to the address the scalability issues and one – affect- N. In MUKD, the server is responsible for the key generation and key distribution process. After authentication, the server generates the TEK and secret keys for each member. Then the server distributes the secret keys to each member in the unicast manner and also securely distributes the TEK by encrypting the TEK using member's SK. The communication cost of proposed MUKD is very low since only one TEK is transmitted during membership changes. Regarding computation cost, MUKD needs one key encryption and one key decryption. Hence the proposed schemes experiences very low computation cost. The amount of keys stored by subgroup manager and members is also considerably reduced in MUKD. The results confirm that the proposed MUKD acquires the good performance in terms of communication, computation and storage costs without compromising the security. The proposed key management scheme is evaluated based on the assumption that the key management path is independent of data path. Hence, it considers the efficiency parameters communication, computation and storage costs. In future, the above schemes can be extended with the combination of key management and data delivery. In addition to these parameters, the proposed schemes can consider other parameters like delay, bandwidth, jitter, etc.

## REFERENCES


1. Abbas Mehdizadeh, Raja Syamsul Azmir Raja Abdullah, Fazirulhisyam Hashim Borhanuddin Mohd Ali, MohamedOthman, Sabira Khatun, 2013. Reliable Key Management and Data Delivery Method in Multicast over Wireless IPv6 Networks. Springer Wireless Pers Communication, 73: 967-991.
2. Abbas mehdizadeh, fazirulhisyam Hasim, Mohamed Othman, 2014. Lightweight decentralized multicast-unicast Key management method in wireless IPv6 networks. Journal of Network and Computer Applications, 42: 59-69.
3. Diffie, W., M.E. Hellman, 1976. New directions in cryptography', IEEE Transactions on information theory, pp: 644-654.
4. Girish Revadigar, Chitra Javali, Weitao Xu, Wen Hu, Sanjay Jha, 2016. Secure key generation and distribution protocol for wearable devices. IEEE International Conference on Pervasive Computing and Communication Workshops, 10.1109/PERCOMW.2016.7457058
5. Junbeom Hur Younho Lee, 2016. A reliable group key management scheme for broadcast encryption. Journal of Communications and Networks, 18: 2.
6. Jian Zhou, Liyan Sun, Xianwei Zhou, Junde Song., 2014. High Performance Group Merging/Splitting Scheme for Group Key Management. Wireless Pers Commun, 75: 1529-1545, DOI 10.1007/s11277-013-1436-x
7. Jia-Lun Tsai, Nai-Wei Lo, 2016. Secure Anonymous Key Distribution Scheme for Smart Grid. IEEE Transactions on Smart Grid, 7: 2.
8. Lein Harn, Changlu Lin, 2014. Efficient group Diffie–Hellman key agreement protocols. Computers and Electrical Engineering (Elsevier)
9. Lin, J.C., K.H. Huang, F. Lai and H.C. Lee, 2009. Secure and efficient group key management with shared key Derivation. Computer Stand Interfaces, 31(1): 192-208.
10. Mittra S Iolus, 1997. A framework for scalable secure multicasting. ACM SIGCOMM Computer communicationReview, 27(4): 288.
11. Nazmul Islam and M.A. Moyeen, 2016. An Empirical Study on Key Management Schemes of Wireless Sensor Network. International Journal of Computer Applications (0975 – 8887) 134: 1.
12. Rafaelli, S., D. Hutchison, 2003. A survey of key management for secure group communication. ACM computing surveys, 35: 309-329.
13. Sherman, A., D. McGrew, 2003. Key establishment in large dynamic groups using One-way Function Trees. IEEE Transaction on Software Engineering, pp: 444-458.
14. Vijayakumar, P., S. Bose, A. Kannan, 2013. Centralized key distribution protocol using the greatest common divisor method. Computers and Mathematics with Applications, 65: 1360-1368.
15. Vinoth Chakkaravarthy, G., Dr.P, Alli, Raja Lavanya, 2016. A cost effective polynomial based multicast unicast key distribution framework for secure group communication in IPv6 multicast networks. IET information security, 10.1049/iet-ifs.2015.0398
16. Vinoth Chakkaravarthy, G. and P. Ambiga, 2014. Dynamic Key Management Schemes: A Survey', Advances in Natural and Applied Sciences, 8(17): 1-7.
17. Vinoth Chakkaravarthy, G., P. Ambiga, T. Grace Shalini, 2015. Energy Efficient Multicast and Unicast Approach for Rekeying in IPV6', International Journal of Applied Engineering Research, ISSN 0973-4562 10: 55.
18. Wang, S., Y. Tsai, C. Shen, P. Chen, 2010. Hierarchical key derivation scheme for group oriented communication Systems. International Journal of Information Technology, Communications and Convergence, pp: 66-76.
19. Wang, W., B. Bhargava, 2005. Key Distribution and Update for Secure inter-group multicast communication. Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks, in conjunction with ACM CCS, pp: 43-52.
20. Wang, W., Y. Wang, 2008. Secure group—based information sharing in mobile Ad Hoc networks. Proceedings IEEE International Conference on Communications,
21. Win Aye and Mohammad Umar Siddiqi, 2006. Key Management for Secure Multicast over IPv6 Wireless Networks. Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking, 6: Article ID 61769
22. Xiaobingb He, Michael Niedermeier, Hermann de Meer, 2013. Dynamic key management in wireless sensor networks : A survey. Journal of Network and Computer Applications, 36: 611-622.
23. Xixiang Lv, Hui Li, Baocang Wang, 2012. Group key agreement for secure group communication in dynamic peer Systems. J.Parallel Distributed Computers, 72: 1195-1200.

24. Yanji Piao, JongUk Kim, Usman Tariq, Manpyo Hong, 2013. Polynomial-based key management for secure intra- group and inter-group communication. Computers and Mathematics with applications, 65: 1300-1309.
25. Young-Hoon Park, Dong-Hyun Je, Min-Ho Park, and Seung-Woo Seo, 2014. Efficient Rekeying Framework for Secure Multicast with Diverse-Subscription-Period Mobile Users', IEEE transactions on mobile computing, 13: 4.