Indian Journal of Science and Technology, Vol 9(14), DOI: 10.17485/ijst/2016/v9i14/87972, April 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Survey on Efficient Group Key Management Schemes in Wireless Networks

Raju Barskar* **and Meenu Chawla**

Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal - 462003, Madhya Pradesh, India; rajubaraskar@rgtu.net, chawlam@manit.ac.in

Abstract

Background/Objectives: With the unequalled growth in Wireless networks, the associated fields of group application have also seen growth like multimedia teleconferencing, stock quoting, and distance education. Security in group applications need to be maintained which cannot be accomplished by wireless networks and IP multicast, encryptions by a shared group key is required. In this paper we review on network-independent group key management, which can be classified into three types: centralized, decentralized and distributed group key management protocols. **Improvements/Methods:** We Analyse key management algorithms in wireless networks and compared various factors of performance like storage overhead, commutation overhead and communication overhead during the join and leave process in groups in wireless group key management approaches, with a set of assessment parameters. In this paper we find various parameters to consider when designing a key management algorithm in wireless networks for mobile environment. **Findings:** It is important to guarantee the safety of this group key and to ensure the group communication. In spite of the fact that group communication encryption can be utilized to secure messages exchanged among group individuals, distributing the cryptographic keys turn into an issue. However, various significant parameters are used to analysis security requirement of application, when the key management algorithm is developed. Strength of key management algorithms is minimization of key cost during the join/leave process of members in the groups. **Applications:** In addition, we identify the relationships between the various security issues of wireless group key management like performance, security and network compatibility with regard to the algorithms discussed. It will give better idea about designing key management algorithms with minimum cost like storage, communication and computation parameters when the members join/leave from the groups. It will enable them to take better decisions.

Keywords: Group Communication, Group Controller, Key Independence, Key Management, KEK, TEK

1. Introduction

In Multicast a message is transmitted from one sender to multiple receivers or from various senders to multiple recipients[1,2]. Without over-burdening the network and resource of the server, multicast empowers the desired applications to benefit numerous clients. Multicast is fevered over multiple unicast if same message is to be sent to various destination. Group communication has numerous difficulties like group privacy and key administration, when a source sends information to a set of recipients in a multicast session. The security of the session is overseen by Group Controller (GC) which is in charge of authentication, authorization and access control furthermore by Key Server (KS) in charge of the maintenance and dissemination of the required key material[3]. IP multicast transmission provides good scalability when implementing group communication due to its open structure. Receivers can join the group and senders can transmit data to a group without any interaction with the central entity. However, this open model lacks any security measures that would enforce access control and protect group communication. In an IP multicast application, any receiver can request data and a receiver does not need to directly contact sender(s) to express its interest in receiving data. Instead, a receiver sends a message to the first multicast-enabled router to register that it is interested in receiving data for a given group. Due to

*Author for correspondence

this anonymous receiving model, the sender is unable to enforce any access control to manage group membership. In some cases, this limitation would render this model unsuitable for commercial situations, as a service provider may prefer to limit content distribution to the subscribers who pay for the service. When IP multicast applications operate in wireless networks, it is more difficult to enforce access control due to the broadcasting nature of wireless networks. Once the group communication data flows into the wireless network, all hosts within the scope of the wireless network can access the data – whether they are members or not, or whether all members pay for the service or just one does. As a result, IP multicast transmission is unable to offer any measures to protect group communication content in wireless networks. Hence, access control needs to be enforced in wireless networks to ensure the security of group communication; this can be achieved through key management.

The multicast security can be classified in to four types: For example receiver access control multicast, authentication of source multicast, finger printing multicast and last one is management of group key multicast. This all types of multicast is used for analyses some security issue in networks. Many researchers were done research work on different types of multicast group security issues. Here we have talk and discuss about group key administration research area.

1.1 Group Key Management in Wireless Networks

In a wireless environment, access control is the most fundamental and critical security issue in group communication[4–6]. For the most part, get to control can be accomplished by applying encryption. To encrypt the group communication information a mutual key, called group key or Traffic Encryption Key (TEK), is utilized and is circulated to all appropriate group individuals. Just the individuals who claim this group key can get to the communication content. The privacy and integrity of the group's communication depend on the safety of the group key. Administration of the group key in this way assumes an important part in the security of group communication[7–10].

Key management in group communication is different from that in the point-to-point communication model. In the point-to-point model, the encryption key can be generated by negotiation through protocols such as the Diffie-Hellman key exchange protocol or it can be generated by one side and then sent to another side[11]. When one side leaves the communication, the connection is automatically terminated, and the encryption key is discarded. Consequently there is no need to update the encryption key. However, in group communication, a group may have many receivers, and the efficient generation, regeneration and delivery of the group key to all receivers is a complicated and challenging task. When one or several members leave the group, the group communication is still active and no one can force the departing member to forget the key. For preventing ex-group members from accessing future communication information, the group key needs to be updated. The group key also needs to be updated whenever a new user joins the group. A new joining member might record the encrypted group communication before it joins the group. In order to decrypt these recorded data, the user joins the group for a short time to obtain the group key. Furthermore, keys for encrypting data should be changed periodically. Cryptographers frown on encrypting a lot of data with the same key because the data is susceptible to the cryptanalysis attack. To sum up, the major task of group key management is to generate, distribute and update the group key to protect the security of group communication. These issues considerably increase the complexity of group key management in wireless networks.

In summary, the major problems and consequent challenges of group key management in the wireless environment are:

- Performance Problems: Group key management in wireless networks needs to provide operational efficiency in communication, computation and key storage to overcome the restrictions of both the wireless network and mobile devices.
- Security Problems: Group key management in wireless networks needs to implement measures to protect the safety of the group key when users join or leave the group. Moreover wireless group key management is required to offer security measures to protect the group key and other supporting keys from being compromised by non-group users or ex-group members[11].
- Network-Compatible Problems: Due to the variety of wireless networks, there is no single group key management approach compatible with all wireless networks. Therefore, a group key management solution needs to be wireless network compatible.

1.2 Group Key Management Requirements

In wide area of group key management schemes which is compatible followings most significant requirements appeared in Figure 1. It shows why we needed in research on group key managements. The most significant fundamentals requirements are classified in to four types: Security requirements, QoS requirements, key server load minimization requirements and group member load minimization[12].

1.2.1 Security Requirements

- Forward Secrecy: In forward secrecy if member of any group leaves the group it should neither able to get any future group key nor be able to decode any group message after departing from the group. This feature also needs to be provided by a secure wireless group key management system[7].
- Backward secrecy: Backward secrecy refers to the prevention of a new member from being able to decrypt the group communication that it has received before it joins the group. A secure wireless group key management system needs to ensure the backward secrecy.
- Collusion Attacks Freedom: A circumstance where any set of leaving members cooperate to recover the present group key by applying the old keying materials known by them is referred as Collusion attack. A secure wireless group key management system should be free from collusion attacks.
- Key Independence: Key independence means that all keying materials should be completely independent from each other. Disclosure of any single key does not compromise other keys. A secure wireless group key management should provide this feature to secure keying materials[8].
- Minimal Trust: Trust relationships refer to that a wireless group key management system should not trust any intermediate or third party components. Should intermediate or third party components be trusted, the effective deployment and operation of the key management approach would be compromised. In general, within a wireless group communication system, the network provider and service provider are different entities. The service provider who owns the group key should not trust the network provider and visa versa.

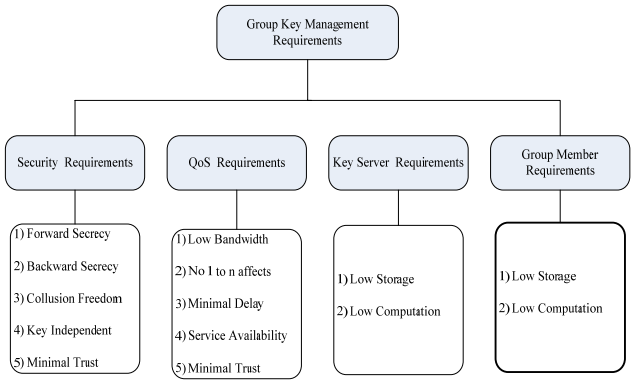

Figure 1. Requirements of group key management in wireless networks.

1.2.2 QoS Requirements

- Low Bandwidth Overhead: It refers to the re-key of the group should not influence due to high number of messages, it applies for change in dynamic groups and it should not limit with group size.
- 1-Affects-N: It refers to the when single membership change during join/leave process, so it applies various members of the group. It save the communication rounds of networks. For example: If single membership changes then it focus another TEK (Traffic Encryption Key).
- Minimal Delays: It refers to minimum delay during the transmission of packet and highly packet delivery during communication when the multicast services are used. Jitters are used to measure the packet delivery ratio. In key management it ply important role, minimization of key changes in key managements because it affects the delays of packet delivery.
- Availability of Services: Service availability the operation of key management structures during the entire multicast session can not be influenced by failure of single node.

1.2.3 Group Member Requirements

- Low Storage: It means minimum number of keys required for communication, so that key servers will be working efficiently and fast access from the memory. In high storage may require more memory and computation for key management.
- Low Computation: If the minimum number for keys is used by key server and group members, so that it required low computation. A benefit of low commutation is increase efficiency and response time of key servers to group members.

2. The Role of Key Management in Group Communication

In communication group key management is highly play important role to achieving secure group communication. The tasks of group key management are to provide: 1. Member identification and authentication, 2. Access control and 3. Management of keying material including the group key and all the supporting keys. Moreover confidentiality, integrity, and authenticity requirements, in a group key management system for dynamic groups the following attributes are necessary[13].

Confidentiality in Groups: In secure group communication, any node from outside the group is not capable to decrypt the messages which are transmitted in the group and all the messages that are encrypted by the session keys established in the group key agreement process.

Scalability: Scalability defines the features of a wireless group key management system which is able to efficiently address the issues of changing group sizes, widely distributed memberships and highly dynamic membership changes. A scalable wireless group key management system is able to deal with group size varying from several dozens to hundreds and thousands. The size of a group should have no or little affect on system performance during key management.

Reliability: Rekeying messages may be lost or delayed because IP Multicast gives best-exertion information delivery just. In the event that a recipient didn't get the rekeying data, it can't decrypt messages which were encrypted with the new group key. All the more genuinely, information might be presented to left members if the sender missed new group key, because the sender still uses old key to encrypt group data. So the framework must give reliable transport of rekeying messages and also provide a recovery system for a member to get missed rekeying messages in a timely manner[15].

Authentication and Identification: Authentication is crucial to secure the system from intruders, who may impersonate members in a legitimate group.

Access Control: Access control provides the access to validate group members and prevents against the unauthorised access control in group communication.

2.1 Procedure of Key Management Schemes

A shared key is applied to encrypt the communication in order to prevent unauthorized users from accessing the content. This encryption key is known as a group key or a Traffic Encryption Key (TEK), and is distributed among all valid group members. The most important issue in group key management is to ensure the safety of keying materials[8]. Three methods are there for group key management, as follows.

- **Key Generation:** Generation of the group key and all other supporting keys is referred as key generation process and helps key distribution controller to distribute the group key to every single genuine receivers.
- **Key Distribution:** Key distribution pertains to the efficient, secure and reliable delivery of keying materials to group members. Because group members may be geographically dispersed or move from one location to another in wireless networks, efficient delivery of the group key to all legitimate members is the most important task in group key management.
- **Key Updating (rekeying):** In key updating process the group key and supporting keys are changed and updated keys are sent to members of group. When the membership is change then group key must be need to updated during the join and leave process in the groups. The key updating is mainly doing for assures the backward and forward secrecy[16].

3. Literature Analysis on Methodology

The group key management schemes can be classified into two main categories: network-independent group key management and network-dependent wireless group key management. Group key management approach takes into account the features of the underlying network architecture when it performs key management. Whereas, the network-independent group key management approaches do not consider the underlying network infrastructure. Network-independent approaches can be applied in both wired and wireless networks as shown in Figure 2, and can be further classified into three types[17]:

3.1 Centralized Group Key Management Schemes

In wireless environment centralized group key management schemes, like as LKH[18] and OFT[19], apply a single KDC (Key Distribution Centre) and a hierarchical key structure to facilitate key management during

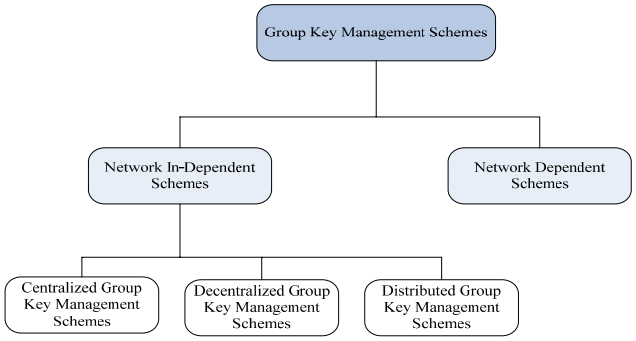

Figure 2. Types of group key management schemes in wireless networks.

key distribution and updating. The centralized approach encounters three major problems: 1. Lack of scalability, 2. Communication and computation inefficiency and 3. Inability to handle multiple-membership changes. In a large and highly dynamic wireless group application, frequent rekeying may overwhelm the capacity of a single KDC and cause failure of key management operations[15]. This failure would jeopardize the security of group application. Moreover, as the number of group users increases, members need to process a larger number of rekeying messages. The frequent keying that result from a large group with high-dynamic membership changes could overwhelm the capacity of lightweight mobile devices. The inability to cope with increasing group size (i.e., the lack of scalability) is the first problem encountered by centralized schemes in wireless networks. The centralized group key management approaches are the best-known and utilized schemes[20–30]. In the next sections, some of important centralized Group Key Management Schemes are discussed in detail.

3.1.1 GKMP (Group Key Management Protocol)

Group Key Management Protocol (GKMP) was proposed in[18], which is used for key agreement in a group. In this mechanism, key server/KDC share a secret key called KEK (Key Encryption Keys) between each valid group members in the groups. The key server/KDC generates a Group Key Packet (GKP) for each valid group members in the groups. This Group Key Packet (GKP) contain two pair of keys: First is Group KEK (GKEK), that is used for encrypt the traffic and next is Group TEK (GTEK), it is used for securely distribution of a new Group Key Packets whenever required. In this mechanism whenever for new member joins in the session, key server needs to generates new key called GKP and forward it to new member in encrypted form by using KEK, now it established as new member, forward its information in encrypted form to other members by using old GTEK. Now key server updated GKP periodically at particular time and it uses the GKEK for distribution GKP to other group members. In other side when a old member want to leaves form the group, then key server needs to generates new GKP, forward it to other members in encrypted form by using KEK, it shall be shared by each members. Thus forward secrecy will be assured. It required O(n) re-key messages when the each leave members from the group. Thus, it is not suitable foe large groups of highly members with dynamic in nature[19].

3.1.2 OFCT (One-Way Function Chain Tree Protocol)

One-Way Function Chain Tree was proposed in[20], which gives same communication overhead with respect to previous approach. In this method, a pseudo-random-generator is used to generate the new Key Encryption Key (KEKs) rather than one-way function and it is applied on those members, which is removal. This proposed method was known as the name of One-Way Function Chain Tree (OFCT)[21]. In OFCT method, pseudo-random-generator is denoted by $G(x)$, where (x) is input and its size is doubles and it contains two functions: $L(x)$ is represent the left halves and $R(x)$ is denoted the right halves. Finally the output of $G(x)$ (i.e., $G(x) = L(x)R(x)$) where $L(x) \vee\vee R(x) \vee\vee x \vee$ Whenever the member u leaves from the group, then the algorithm to rekey of the tree is performed according to the manner as discussed below:

- For the first node, every node v from u to the root of the tree is correlated to the new generated value r_v with by using $r_{v(u)} = r$ and all other node of v it will be $r_{p(v)} = R(r_v)$ (where $p(v)$ denotes the parent of v).
- $k'_v = L(r_v)$ is newly generated key, which is used.
- For all other nodes $r_{p(v)}$ is encrypted by using key $k_{s(v)}$ (where $s(v)$ called the sibling of v).

3.1.3 HACT Protocol

A new technique for group communication was proposed in[22]. According to this method clustering of members is performed around single leaves of a *a-arytree* proposed[23], it also show how to computer size of cluster using Cantti's model. Here in the group separating n member into m size

of m clusters. Hence the complexity of depth of the tree is $log_a(n/m)$. In this technique KDC shares unique key k_i which is assigned to every member of cluster, this unique key is shares among members of each and every cluster by using with help of KEK. Thus, KDC uses random number called r, as index, it is useful for pseudorandom function f_r to produce the key k_i for members i ($k_i = f_r(i)$) Therefore, every members of cluster can only carries to know random number called r, as index and the cluster KEK. Now cluster members which is belongs to same cluster needs to shares a set of keys with the path from the leaf node to the root, for that reason the all members which is belong to same cluster must be holds also $log_a(n/m) + 1$ KEKs. Hence if the member leaves from the cluster is immediate receive a new cluster KEK. This new cluster key is already encrypted by all individual KEKs of remaining members. Therefore it only required m-1 encryptions whenever member is leaves from cluster and needs to for revise all remaining members common cluster key KEK. Apart from that KDC updates its new KEK with old respective node's KEK for assured forward secrecy and safe communication, this procedure will follow from path from the cluster leaf to the root. Hence, when a single member is deleted, the update message has $m - 1log_a(n/m)$ keys.

3.1.4 CFT (Centralized Flat Table Protocol)

Centralized Flat table Key Management was proposed in[14]. In this method, flat table is used by key server for maintaining keys rather than hierarchical key tree, in order to reduce the number of keys maintained by the Key sever. This flat table consists of two different keys one of Traffic Encryption Key (TEK) and another is $2w$ Key Encryption keys (KEKs), where w denoted the number of bits in the member *id* These two keys KEK assign for each bit in the member id, one associated with each possible value of the bit 0 or 1. Each group member w keys associated with the state of its bit of its member id. Illustrated the structure of CFKM for 4 bit *id*

Table 1. Centralized flat table with $w = 4$

	TEK	
ID bit #0	KEK_{00}	KEK_{01}
ID bit #1	KEK_{10}	KEK_{11}
ID bit #2	KEK_{20}	KEK_{21}
ID bit #3	KEK_{30}	KEK_{31}
	Bit value = 0	Bit value = 0

Every legal member brings the KEKs which are linked with the identifier bits value. Therefore, each legal member holds keys (KEKs with TEK). Such as, legal members who associate with the identifier 0101, it keeps KEK00, KEK11, KEK20, KEK31 and also the TEK. In this mechanism, whenever a legal member is leaves from the group, then all keys related to this departing legal member shall be updated for the reason of forward secrecy. Hence, KDC sends a rekeying message, which separated into two parts: In First part holds the TEK encrypted form by using with each not correlated KEK from the flat table. In order that subsequently all the other members would be capable to decrypt the new TEK. In next remaining part holds the new KEKs in encrypted form by help of the both old KEK and new KEK. Therefore, it is ensure that leaving legal member not cable of recover new TEK and the all other remaining legal members can modifies its old KEKs beyond any access to the KEKs of the other members[25].

3.1.5 LKH (Logical Key Hierarchy Protocol)

The Logical Key Hierarchy (LKH) approach has been proposed independently in[24,25]. This was the one of the most prominent and systematic group key management algorithms. The major contribution of LKH is to apply a hierarchical structure to facilitate group key management. Theoretically, any hierarchical tree structure can be applied in LKH. Without loss of generality, apply a binary tree as an example to illustrate LKH, due to the simple establishment, management and node operations of a binary tree. In a LKH key tree, the internal nodes of the tree hold supporting keys (Key Encryption Keys KEK, that is used to encrypt the group key for distribution to the group members. Each one leaf node is correlated with a group member and holds a pair-wise KEK. The KEK of leaf node is determined when a new user joins the group and is only known to the joining member and the KDC. Each member needs to keep a set of KEKs along the path from its leaf node to the root node. For a balanced binary tree, each member stores h +1 key, where h is the height of the tree and equals $log2n$, and n is the number of group members.

3.1.6 ELK (Efficient Large-Group Key Protocol)

ELK is known as Efficient Large-Group Key protocol proposed in[26]. The ELK protocol is very similar to the OFT in which a parent node key is generated from its children keys and this method based on hierarchical tree.

In the hierarchical tree for constructing and manipulating the keys, this protocol uses PRFs known as pseudo-random functions. Here pseudo-random functions are used to generate output with length of *n* by the key K of input M with length of *m*. The following notation is used to represent for calculation: $PRF_k^{m\to n}(M)$. By using pseudo-random functions on a key, it is easy to generate four types of different keys with different contexts: $K_i^{\alpha} = PRF_{k_i}^{n\to n}(1)$ it used to produce values *n1* and *n2*, $K_i^{\beta} = PRF_{k_i}^{n\to n}(2)\, ki$ (2), second function is used encrypt key update messages, $K_i^{\gamma} = PRF_{k_i}^{n\to n}(3)$, third function is used to produce hints, and $k\pm I\ DK_i^{\delta} = PRF_{k_i}^{n\to n}(4)$, used to update key nodes. ELK method is more efficiently for rekeying production in timely, so that it much better to updated key tree completely within the given time interval. The group key of ELK is updated using the derivation $k0\ G\ DPRF\ K'_G = PRF_{k_G^{\gamma}}^{n\to n}(0)$ (group key) and all other $k0$ *i* are derived by $k0\ I\ DPRF\ K'_G = PRF_{k_G^{\gamma}}^{n\to n}(K_G)$. It is also updated new driving key to other during the join operation, ELK does not supported multicast messages.

3.1.7 OFT(One-way Function Tree Protocol)

The OFT known as One-way Function Tree has been proposed in[27]. The important contribution of OFT against LKH is that, for minimization of computation and communication cost, it permits all members of group can compute key locally.

OFT applies the same tree structure as LKH to manage keys. The root key serves as a group key and every member is related with a unique leaf node and knows a set of KEKs from its leaf node to the root[30]. When a member joins a full and balanced binary tree with *n* members, in a one way function tree scheme, the KDC needs to send the new member *h* blinded keys, where *h* denotes the height of the key tree equalling 2*n*. Further, the KDC needs to send *h* blinded keys to the current members. Thus, the KDC needs to encrypt 2 *h* blinded keys and send *h* + 1 rekeying messages for a join action. The KDC needs to re-calculate *h* blinded keys and sends *h* rekeying messages when a member leaves the group, where *h* is the height of the key tree for OFT

Advantages: In CGKM schemes, straight forward management model is used. Thus it easily calculate communication computation and key storage cost by using logarithmical level of communication computation and key storage cost. It is easy for implementation.

Limitations: Scalability issue makes it unsuitable for large and highly-dynamic wireless group applications. During Rekeying process communication inefficiency among the groups when it dealing with multiple-memberships.

Table 2. Comparison complexity of various centralized group key management schemes

Schemes	1-affects-n	Re-key Overhead			Storage Overhead	
		Join		Leave	KDC	Member
		Multicast	Unicast			
Simple[15]	No	n	0	n	n	0
GKMP[18-19]	Yes	2	2	$2n$	$n+2$	3
OFCT[20-21]	Yes	$log_2 n$	$log_2(n)+1$	$log_2(n)+1$	$2n-1$	$log_2(n)+1$
HATC[22-23]	Yes	$m-1$ $+log_a(n/m)$	$log_a\left(\frac{n}{m}\right)_2$	$m-1$ $alog_a\left(\frac{n}{m}\right)$	n/m	$log_a\left(\frac{n}{m}\right)+2$
CFT[14]	Yes	$2I$	$I+1$	$2I$	$2I+1$	$I+1$
LKH[24-25]	Yes	$log_2(n)+1$	$log_2(n)+1$	$2log_2(n)$	$2n-1$	$log_2(n)+1$
ELK[26]	Yes	0	$log_2(n)+1$	$log_2(n_1+n_2)$	$2n-1$	$log_2(n)+1$
OFT[27]	Yes	$log_2(n)+1$	$log_2(n)+1$	$log_2(n)+1$	$2n-1$	$log_2(n)+1$
Secure Lock	No	0	2	0	$2n$	2

Table 3. Performance of security parameters among related works

Features → / Key Management Techniques ↓	Backward Secrecy	Forward Secrecy	Secure Against Collision	Anti-eavesdrop	Data integrity	Verifiability	Repudiation	Privacy
Simple[15]	Y	Y	Y	N	Y	N	Y	Y
GKMP[18-19]	Y	Y	Y	N	Y	Y	Y	Y
OFCT[20-21]	Y	Y	N	Y	N	Y	Y	Y
HATC[22-23]	Y	Y	Y	Y	N	Y	Y	Y
CFT[11]	Y	Y	N	N	Y	Y	Y	Y
LKH[24-25]	Y	Y	Y	Y	Y	Y	Y	Y
ELK[26]	Y	Y	Y	Y	Y	Y	Y	Y
OFT[27]	Y	Y	Y	Y	Y	Y	Y	Y

3.2 Decentralized Group Key Management Architecture

Decentralized architecture provides a useful approach to tackle the scalability for group key management in a large area by dividing the whole group into several small subgroups. These schemes are suitable to support group key management for large-scale wireless networks such as cellular wireless network, WiMax[49,50] and the future 4G systems[51]. Furthermore, decentralized architecture is the only solution to address the 1-affect-N phenomenon that also needs to be considered when dealing with wireless networks. However, decentralized architecture only proposed a framework for large-scale group key management; it does not provide an approach for efficiently distributing keying materials to group members in subgroups. Therefore, decentralized architecture needs to cooperate with the other group key management approaches to provide an integrated solution for group key management in wireless networks. Furthermore, in wireless networks, third-party entities are commonly involved in decentralized architectures. The wireless network operator and secure group application providers are generally different entities. Establishing a trust relationship between them is a critical security concern that also needs to be addressed by wireless group key management approaches. Several decentralized group key management approaches can be classified into this type, including[28–36].

3.2.1 SMKD(Scalable Multicast Key Distribution

SMKD which is known as Scalable Multicast Key Distribution proposed in[28] RFC1949 in 1996. Further extended this protocol by using tree build, it is well known as Core Based Tree multicast routing protocol (CBT) and used to distribute keys to the group members in multicast manner. CBT architecture is divided into two cores: In first main core, in this core the multicast tree are rooted and secondary core, it used for validation exist ultimately[29]. The main core is used to create an ACL (Access Control List). Here GTEK (Group Traffic Encryption Key) and GKEK (Group Key Encryption Key) is used for updating the session key of group. In ACL, the GTEK and GKEK is further pass to the secondary cores and other node only join the multicast tree group after the their authentication. For nay node those want to join the group is authenticated by secondary core and this authentication will be followed by primary core to distribute the keys with ACL, but here only the main core is generates those keys. In SMKD, not show the explanation against security aspect like forward secrecy, unlike than not recreate a new group without the leaving members of group.

3.2.2 Iolus Protocol

Lolus introduced a framework for scalable secure multicasting that uses a hierarchy of subgroups for performing group key management to address the scalability[30]. In this framework, the large group is decomposed into a number of subgroups which form a tree structure. Each subgroup has a controller called Group Security Intermediate (GSI) or as a Group Security Agent (GSA). The GSIs in the top level of the tree are managed by a Group Security Controller (GSC). Each subgroup has its own subgroup

key, a GSI builds a connection channel between its parent's subgroup and its own subgroup, so a GSI is a member of its parent's subgroup as well as member of its own subgroup. A GSI thus owns at least two subgroup keys. Even though Iolus is scalable, it has the drawback of affecting the data path. When a sender sends a message to the group, it encrypts the message with its subgroup key and multicasts it within the subgroup. Once a GSI receives the message, it decrypts the message with the key of the current subgroup, and then encrypts the message again with its other subgroup key and forwards this message to the other subgroup.

3.2.3 DEP: Dual Encryption Protocol

DEP was proposed in[31], commonly known as the Dual Encryption Protocol. In this protocol, they suggested SGM known as sub-group manager for hierarchical sub-grouping of the group members and control each sub-group. Now the Key Server I s shares KEK_{i3} with SGM_i. For broadcast the DEK to the group members, a key server generates and transmits the packaging message of DEK with encrypted with KEK_{i2} and encrypted again with the KEK_{i3}. Whenever receiving the packaging message by key server, now SGM_i can decrypts again its remaining parts of the message using KEK_{i3} and find out the DEK encrypted with its sub-group KEK (KEK_{i2}), which is not called by the SGM_i. SGM_i encrypts the encrypted DEK using KEK_{i1} forward it to the other sub-group members and also sends it out to sub-group i . Every members in the sub-groups i can be able to decrypts this message using KEK_{i1} and then, by decrypting the message using KEK_{i2} (shared with KS), recovers DEK. Thus, DEK can not be re-calculated whenever the both keys are known by member. In this mechanism third party of management is involve in the form of (SGM_s), they always safe the key and do not have access to the group key (DEK). Whenever membership is changed of a member in a sub-group i then only the SGM_i changes KEK_{i1} and broadcast to the all members. In Future DEK changes due to membership change, it not able to access for members of sub-group i that does not received the new KEK_{i1}.

3.2.4 STB Protocol

In STB mechanism number of trusted routers is used. This routers and receiver contain their public and private keys pairs. The public keys pair will generate from public key or it is necessary to know at least one public key of its neighbours. Whenever sender want to send message, first it choose random session key , that is called DEK (Data Encryption Key), and encrypts this message by using DEK and first router public key. After receiving a message, a router decrypts this message by using DEK and again re-encrypts by using the DEK and public keys of the neighbouring routers/hosts. This encrypted data packet is not modified by any intruder. Now further new packet is forwarded to another routers/hosts. This procedure is repeated whenever the data packet is forwarded to receivers. Thus receiver receives this data packet and decrypt by using DEK and its private key and again message can decrypt using DEK. The main objective of STB mechanism is to provide general solution for unicast and multicast for message transmission. Therefore, here there is no need for an individual multicast group to maintain its own keys, and thus the key management problem is solved naturally[32].

3.2.5 MARKS

The MARKS plan was proposed in[33] which characterizes steady overhead key distribution protocol. MARKS is based upon the reason that numerous applications, e.g., subscription pay-TV, pre-paid or or pay-per-view, don't require untimely eviction. Subsequently, the protocols consider that the term over which a member stays in the group is known when the member joins. For single changes in the group, without the known from the earlier participation duration necessity MARKS plan gives better execution limits. Every meddler node takes into account producing two offspring's' i.e., left and right children's. To begin with left middle node is produced, then applying the building functions on the primary node and then moved it to the parent one bit to the left. Before applying the building function on the second hub, second right intermediary node is produced. To access the group communication user must be create the required keys, it implies unauthorized user can't get to. It not required changing framework, since the keys are changed as a reason for the time at point when membership required changing group key. The dispersion of the seeds and the administration of receivers' inquiries are guaranteed with an arrangement of key managers.

3.2.6 Cipher Sequence

The CS is popularly known as Cipher Sequences is a framework based on reversible cipher sequences used for

multicast security and proposed in[34]. A function $f(S, a)$ is known as Cipher Group (CG), if it exist the following properties and characteristics: A sequence of n elements i.e. $a_{i}^{1 \le i \le n}$), and there is a sequence of $n + 1$ elements $S_i (0 \le i \le n)$ such as $S_i = f(S_i - 1, a_i)$ for $i > 0$ and S_0 is the initial value; and for every pair $(i, j)$where $i > j$, there exists a function h_{ij} such as $S_i = h_i, j(S_j)$ At the end of source, multicast tree is in rooted form, if the members are leave from group then internal nodes elements are immediate change for multicast communication.

Now, consider S_0 know the message which to be multicast and in this section every node N_i be allocated a values $a_i > 1$. Whenever the node N_i getting a value S_j from its parent N_j, it generates $S_i = f(S_j, a_i)$ and forwards it to S_i to its children that may be any leaves or other internal nodes. Those who are leaves are assigning the function $h_{0,n}$,which cable of them to calculate S_0 from S_n, since $S_0 = h_{0,n'},(S_n)$, and therefore re-generate the original message or data.

For this instance Molva's scheme are used that is explain as follows:

- The value $S_1 = f(S_0, a_1)$ is calculated by root and forwards S_1 to N_2, and N_4.
- Now the value $S_2 = f(S_1, a_2)$ is calculate by Node N_2 and forwards S_2 to N_3.
- Next section the value $S_3 = f(S_2, a_3)$ is calculated by Node N_3 and forwards S_3 to leaves L_4.
- Leaves L_4 calculate $S_0 = h_0, 4(S_3)$ and re-generate the original data:S_0.

3.2.7 KRONOS

The Kronos protocol was proposed in[35]. In this protocol, the group rekeying takes place at a fixed interval, not by membership change. The framework of Kronos is similar to IGKMP but, there exist two major difference between them. First, the DKD is not directly involved in producing MKey (traffic key). Instead of each AKD independently generates the same group wide traffic encryption key at a fixed interval and multicasts it to the group members of its sub group. Second, the periodic rekeying causes the decoupling of the frequency of rekeying from group size and dynamic membership change. There are two factors that need to be addressed for Kronos to work properly. First, all AKDs should generate same traffic encryption key (MKey) synchronously by using clock synchronization protocol and Network Time Protocol (NTP). Second factor is that all AKDs must share some state information and the same key generation algorithm. So, they can generate the same key at anytime. Kronos does not use the group controller or sub group controller for generating the group key independently. Consequently, the networks become fault-tolerant. But Kronos generates new group key based on the previous one. Hence, if one key is disclosed then all the following keys are compromised. This finally consider as the bottleneck problem of Kronos.

3.2.8 IGKMP

The IGKMP is the two-level Intra-Domain Group key management protocol, which was proposed in[36]. In IGKMP, (i.e., as in Iolus) the group key management domain is divided into a number of small areas. A member resides in one and only one of these small areas. The two types of Key Distributors (KD) are deployed in IGKMP.

- At the group domain level, a Domain Key Distributor (DKD) is defined for the purpose of key management within the domain.
- At the area level, an Area Key Distributor (AKD) is defined for key distribution within the area.

All the AKDs form a multicast group referred as the All-KD-Group, which is used by the DKD to transmit rekeying messages to all AKDs within the domain. When a membership change occurs, the DKD generates a new group key, and sends it to all the AKDs via the All-KD-Group. Once the AKD receives the new key, it distributes this new group key within its own area. The AKD communicates with members in its area either through a secure channel or through multicast transmission.

3.2.9 HYDRA

The Hydra protocol was proposed in[37]. In this protocol a large group is divided in to smaller sub groups, and key server called as Hydra Server (HS). Hydra is a decentralized group key management scheme there is no central subgroup controller. Hydra architecture is divided into two most important hierarchical levels. The top level is associated with the Hydra Servers and the bottom level is associated with the members, separated by subgroups. A Group Key, is shared by all the members in the group to secure the group communication. The Hydra Key is shared by all the HS_s for communicating the HS group. Each sub group has a subgroup key, shared by HS and all subgroup members, to protect the Group Key within

a subgroup in the Hydra- group. When a new Group Key is distributed to all the HS for controlling the group, the HSs encrypt the new group key with their corresponding sub - group key and then send it out to all the members. Hydra solve the bottleneck problem of Iolus, IGKMP and Kronos. Hydra does not use centralised controller for group key generation, and it is able to avoid single point of failure.

Advantages: The DGKM architecture provides a framework to address Group Key Management for large-scale wireless networks. It is also capable of to address the 1-affect-N phenomenon.

Limitations: In DGKM needs to cooperate with other Group Key Management approaches to form a comprehensive solution. It also required third party trust relationship and cannot work alone.

3.3 Distributed Group Key Management Approach

Distributed group key management schemes abolish group controllers to avoid the possibility of single-point errors, which are associated with centralized group key management approaches. As an alternative distributed group key management approaches provide fault-tolerance by using member contributions to generate group key. Distributed group key management approaches allow all group members to compute the same group key independently and locally. Any single member failure can be ignored, because it does not prevent the other members from reaching a shared group key. However, this fault-tolerance feature sacrifices operational efficiency at the cost of communication and computation. While the Diffie-Hellman key exchange protocol[12] is widely applied in distributed schemes in order to derive a group key, its expensive exponentiation computation may not be affordable to mobile devices. Furthermore, it takes a long time for all group members to reach a group key. Along with the growth in the group size, the convergence time of group key generation increases. There are several distributed group key management schemes have been proposed, including the[38–48] is widely applied in these approaches.

3.3.1 Ingemarson Protocol

Ingemarson Protocol was proposed in[38], which is one of the most basic proposals for group communication to extend Diffie-Hellman group key agreement protocol[12]. In this mechanism all members are organized into virtual ring, for example member M_i can be communicates with M_{i+1} member of group and member M_n can communicate with member M_1. To generate key for group it required $(n-1)$ rounds by the group members. Firstly, for generation of group key each member M_i required random value N_i, computes g^{N_i} and forwards it to the another memberM_{i+1}. Then each member M_i required in each round to the power of N_i and intermediate value added to the member M_{i-1}, and forward to the result to the M_{i-1}. In this round each member required n exponentiations gets the group key $K_n = g^{N_1 N_2 \ldots\ldots N_n}$ after $(n-1)$ rounds. The drawback of this mechanism is that it is not appropriate for dynamic groups as it require more execution time for entire algorithm after each membership change.

Table 4. Comparison of various factors in decentralized group key management schemes

Key Management Schemes	Key Independent	1-affacts –n	Local Re-key	Communication Types	Fault Tolerant	Decentralized		Rekey
						Mngmt	KDC	
CBT[28-29]	Y	N	N	BOTH	N	Y	Y	N
Iolus [30]	Y	Y	Y	1-to n	Y	Y	Y	Y
DEP[31]	Y	N	N	BOTH	N	Y	N	Y
STB[32]	Y	Y	N	BOTH	N	Y	Y	N
MARKS[33]	N	N	N	BOTH	Y	Y	-	N
CS[34]	Y	Y	N	1-to-n	N	N	N	Y
Kronos[35]	N	N	N	BOTH	Y	Y	Y	N
IGKMP[36]	Y	N	N	BOTH	N	Y	N	Y
Hydra[37]	Y	N	N	BOTH	Y	Y	Y	Y

3.3.2 GDH.3

GDH.3 was proposed in[39], three versions of Group DH Key Exchange scheme (GDH). GDH extends the two-party Diffie-Hellman key exchange protocol into a group operation. In GDH.1 and GDH.2, the overhead of computation is quite considerable due to the large number of exponentiation calculations. In order to decrease the number of exponentiation computations, GDH.3 was proposed, in which it requires to perform a constant small number of exponentiation computations by each member only. GDH.3 thus offers a reduced computation cost compared to the previous two versions. There are four steps in GDH.3.

Step 1: The first step has $n - 2$ rounds and is used to collect the members' contribution from member u_1, u_2, ... u_{n-1}.

Step 2: In the second step, member u_{n-1} broadcasts value $\alpha^{S_1 S_2 \ldots\ldots S_{n-1}}$ to all the members.

Step 3: In third step, every member extracts its own component, s_i from the value $\alpha^{S_1 S_2 \ldots\ldots S_{n-1}}$, and sends the result to the last member u_n.

Step 4: Finally, member u_n raises all the received values to its secretcomponent s_n and broadcasts these results to the members. After receiving the values from member u_n, every member can compute the group key by raising the value to the power of its own component. A problem of GDH.3 is that the last member in the group is a special user whose performance determines the failure or success of the group key generation. In GDH.3, the last member of the group receives n messages and it required n exponentiations computations for the purpose. This mechanism every member have penalty in the form of storage space and strong computational power.

3.3.3 CKA

The CKA is known as Conference Key Agreement which was proposed in[40]. In this key agreement scheme, each and every group members contribute to generating the group key and it is generated with the help of combining function: K = f(N_1, $h(N_1)$,...,$h(N_n)$) where f called as combining function (a MAC), h denote here one-way function, n is size of group, and N_i is the parts of the group member i. The protocol define that $n - 1$ members for broadcast their contributions (N_i) in the communication. For example if group leader U_1, encrypts its contribution (N_1) by using the public key of each member and transmit it. Then all remaining group members can easily encrypt N_1 can decrypt it by using public key and generate the group key.

3.3.4 Octopus Protocol

The Octopus protocol was proposed in[41]. This is based on Diffie-Hellman key exchange protocol[42]. In Octopus protocol, the big group is comprises of n members and splitted into four sub-groups $by\,(\frac{n}{4}\,\text{membereach})$. In their every sub-group agree with an intermediate DH value: $I_{subgroup} = \alpha^{u_1 u_1 \ldots\ldots u_{n/4}}$, here u_i denoted the contribution from user i, then this intermediary values is exchange by the subgroups. All group members needed to calculate the group key. Here leader are responsible for contributions value of sub-group members and also responsible for calculating of the intermediary DH value ($I_{subgroup}$). Thus, in there group leaders denoted A, B, C and D. First two leaders A and B can exchange their intermediary values I_a and I_b using DH and creating $\alpha^{I_a \cdot I_b}$. Also such type second two leader C and D can exchange their intermediary values I_c and I_d and creating $\alpha^{I_c \cdot I_d}$. Then also after exchange A and C calculate $\alpha^{I_a \cdot I_b}$ and $\alpha^{I_c \cdot I_d}$. Also after doing same thing with leader B and D. Now, all of them can calculate $\alpha^{I_a \cdot I_b \cdot I_c \cdot I_d}$. After that, A, B, C and D send to their respective subgroups $\alpha^{\frac{\alpha^{I_a \cdot I_b \cdot I_c \cdot I_d}}{u_i}}$, where i = 1...(n – 4)/4, and all members of the group are capable of calculating the group key.

3.3.5 DOWFT

DOWFT is another distributed fashion approach based on logical key hierarchy and was proposed in[42]. This protocol overcome the drawback of one-way function tree proposed in[26]. There is no centralized controlling system in logical key hierarchy. Therefore, every member in a group, which is involve in group communication are trusted with access control and key generation. Every member belongs to particular group is responsible for generating, own key and sending the blinded version of this key to its sibling. As in the centralized one-way function tree, every member knows all keys in the path from its node to the root node and all blinded keys from the sibling node to the nodes in the path to the root.

3.3.6 DFT

DFT known as Distributed Flat Table, which was proposed in[14]. In this mechanism flat table is uses in distributed fashion and there is no requirement Group Controller. In distributed flat table scheme, member knows only KEKs that it is entitled to. At any time, no members know all the keys. The distributed management has problem, such as namely that a joining member is obliged to contact a group of members to get all the keys needed. Moreover, while many members could be changing the same key at the same time, there could be serious delays in synchronizing the keys.

3.3.7 DH-DLKH

A new distributed approach which is based on the logical key hierarchy, introduced in[43]. In this hierarchy approach, the GC (Group Controller) is completely eliminated and generated logical key hierarchy among all the members in group, thus there is no entity that knows all the keys at the same time. This protocol uses the notion of sub trees agreeing on a mutual key. There are two groups and members of groups namely sub tree L and sub tree R, agree on a mutual encryption key. Member m_l is assumed to be leader of left subtree and member m_r is again leader of right subtree. The key of left subtree is k_L and next key of right sub tree are k_R. This mechanism works as follows, when it agree on a mutual key[44].

- Member m_l chooses a new key k_{LR}, and sends it to member m_r using a secure channel.
- Member m_l encrypts key k_{LR} with key k_l and multicasts it to members of subtree L; member m_r encrypts key k_{LR} with key k_R and multicasts it to members of subtree R.
- All members (LUR) receive the new key.

3.3.8 STR

Skinny Tree (STR) protocol was proposed in[45]. This is a contributively protocol that use a tree structure. The leaves are associated with the group members and each leaf is well-known by its position LN_i in the tree and holds a secret random r_i (generated by the corresponding member M_i) and its public blinded version $br_i = gr_i modp$, where g and p are DH parameters. Each internal node is well-known by its position g_i in the tree and holds a secret random IN_i and its blinded public version $bk_i = g^{ri} modp$. Every secret k_i is recursively computed by the formula as follows: $k_i = (bk_{i-1})^{r_i} \ modp = (br_i)^{k_{i-1}} \ modp$. The group key is the key associated to the root: $k_n = g^{r_n} g^{r_{n-1}} \ldots\ldots g^{r_2 r_1}$ This induces a $0(n)$ key calculations in order to establish the group key associated to the root of the tree, due to the linear structure of the tree. In addition, each member must store and maintain all the public keys associated to all the nodes of the tree. In case of a membership change (join/leave) the tree is re-built consequently and all the

Table 5. Comparison of complexity in distributed key management protocol schemes

Schemes	No. of Rounds	No. of Messages		DH Exchange	Leader Required
		Multicast	Unicast		
Ingnemarson et al.[38]	$n-1$	0	$n(n-1)$	Yes	No
GDH.3 [39]	n	n	$n-1$	Yes	No
Conference Key Agreement [40]	3	n	$n-1$	No	Yes
Octopus Protocol [41]	$2^{(n-1)}/4+2$	0	$3n-4$	Yes	Yes
Distributed One-way Function Tree[42]	$log_2 n$	0	$2log_2 n$	No	No
Distributed Flat Table[14]	n	0	$2n-1$	No	Yes
Distributed Logical Key Hierarchy[43-44]	3	1	n	No	Yes
DH-Distributed Logical Key Hierarchy[45]	$log_2 n$	$log_2 n$	0	Yes	No
STR protocol[45-46]	n	n	0	Yes	No
Tree-based Group DH Key Management (TGDH)[47]	n	n	n	Yes	No

members update the group key which is the new key k_n associated to the root of the tree[46].

3.3.9 TGDH

Tree based Group Diffie-Hellman key management (TGDH) was proposed in[47] to extend the two-party DH protocol to a hierarchical structure to secure multi-party communication. In TGDH, each and every member construct an identical virtual binary tree that may or may not be balanced. Each member is associated with a leaf node in the key tree. Instead of applying a one-way function to generate the keys in the upper level, members use Diffie-Hellman protocol to generate the keys along the path from its leaf node to the root[12]. When a user, u, wishes to join the group, it broadcasts its public shared blinded key to the entire group. Next, all current members determine the insertion location for u on the tree and the sponsor of u. Each member updates the maintained key tree by adding a new leaf node for u and a new internal node, and removes all secret keys and blinded keys from the sponsor's leaf node to the root node. Then, the sponsor generates its new secret key and calculate all blinded keys from its leaf node to the root, and broadcasts all these new blinded keys to the whole group. Every member is able to calculate the group key after receiving these new blinded keys.

Advantages: Distributed group key management schemes are used for fault-tolerance to avoid single-point error.

Limitations: The DGKM schemes are expensive not affordable to mobile devices and not suitable for wireless networks. These approaches are not suitable for large and highly dynamic secure group applications.

4. Conclusion

In wireless network, group key management is vital for group applications to ensure the security and to provide secure communication with the help of shared group keys. In this paper we concluded different key management solutions for efficiency and security problems associated with group applications in wireless networks. We discussed the network dependent and network independent techniques for group communication. Main focus was on network-independent approaches. The three categories of Network-independent schemes are as follows: distributed, centralized and decentralized group key management. Group key management methodologies and reduce the communication, computation cost, key independence, local rekey, fault-tolerance to avoid single-point error and key storage overhead for both the KDC and members. Major challenges are scalability, the 1-affect-N phenomenon and the trust relationship issue. We need to focus on these challenges of centralized, decentralized and distributed group key management schemes in wireless networks.

5. References

1. Rajkumar K. Efficient resource allocation in multicasting over mobile adhoc networks. Indian Journal of Science and Technology. 2014 Jun; 7(S5). DOI: 10.17485/ijst/2014/v7iS5/50711.
2. Toghian M, Morogan MC. Suggesting a method to improve encryption key management in wireless sensor networks. Indian Journal of Science and Technology. 2015 Aug; 8(18). DOI: 10.17485/ijst/2015/v8i19/75986.
3. Challal Y, Seba H. Group key management protocols: A novel taxonomy. International Journal of Information Technology. 2005; 2(1):105–18.
4. Gong L, Shacham N. Multicast security and its extension to a mobile environment. Wireless Networks ACM-Baltzer Journal. 1995; 1(3):281–95.
5. Vyas P, Trivedi B, Patel A. A survey on recently proposed key exchange protocols for mobile environment. Indian Journal of Science and Technology. 2015; 8(30). *DOI: 10.17485/ijst/2015/v8i30/72068.*
6. McHugh J, Michael JB. Secure group management in large distributed systems: What is a group and what does it do. Proceedings of the ACM Workshop on New Security Paradigms; 1999. p. 80–5.
7. Bruschi D, Rosti E, "Secure Multicast in Wireless Networks of Mobile Hosts: Protocols and Issues", Mobile Networks and Applications, Springer journal, 2002 Dec, 7(6), pp. 503-511.
8. Hardjono T, Tsudik G. IP multicast security: Issues and directions. In Annales des telecommunications. Springer Journal. 2000; 55(7-8):324–40.
9. Hong S. Multi-factor user authentication on group communication. Indian Journal of Science and Technology. 2015 Jul; 8(15). DOI: 10.17485/ijst/2015/v8i15/72941.
10. Moyer MJ, Rao JR, Rohatgi P. A survey of security issues in multicast communications. IEEE Network Journal. 1999; 13(6):12–23.
11. Diffie W, Hellman ME. New directions in cryptography. IEEE Transactions on Information Theory. 1976; 22(6):644–54.
12. Devi SD, Padmavathi G. Secure multicast key distribution for mobile ad hoc networks. arXiv preprint arXiv; 2010. p. 218–23.

13. Harney H, Colgrove A, McDaniel P. Principles of policy in secure groups. Proceedings of the Network and Distributed Systems Security Internet Society; 2001. p. 125–35.
14. Waldvogel M, Caronni G, Sun D, Weiler N, Plattner B. The VersaKey framework: Versatile group key management. IEEE Journal on Selected Areas in Communications. 1999; 17(9):1614–31.
15. Khan AS, Fisal N, Bakar ZA, Salawu N, Maqbool W, Ullah R, Safdar H. Secure authentication and key management protocols for mobile multihop WiMAX networks. Indian Journal of Science and Technology. 2014; 7(3):282–95.
16. Amir Y, Kim Y, Nita-Rotaru C, Schultz JL, Stanton J, Tsudik G. Secure group communication using robust contributory key agreement. IEEE Transactions on Parallel and Distributed Systems. 2004; 15(5):468–80.
17. Rafaeli S, Hutchison D. A survey of key management for secure group communication. ACM Computing Surveys (CSUR). 2003; 35(3):309–29.
18. Harney H, Muckenhirn C. Group Key Management Protocol (GKMP) specification. RFC 2093; 1997.
19. Harney H, Muckenhirn C. Group Key Management Protocol (GKMP) architecture. RFC 2094; 1997.
20. Canetti R, Garay J, Itkis G, Micciancio D, Naor M, Pinkas B. Multicast security: A taxonomy and some efficient constructions. Proceedings, Computer and Communications Societies, 18th Annual Joint Conference of the IEEE; 1999. p. 708–16.
21. Goldreich O, Goldwasser S, Micali S. How to construct random functions. JACM. 1986; 33(4):792–807.
22. Canetti R, Malkin T, Nissim K. Efficient communication-storage tradeoffs for multicast encryption. Advances in Cryptology-EUROCRYPT'99. Springer Berlin Heidelberg; 1999. p. 459–74.
23. Li M, Poovendran R, Berenstein C. Optimization of key storage for secure. Proceedings of the 35th Annual Conference on Information Sciences and Systems (CISS); 2001.
24. Kei WC, Gouda M, Lam SS. Secure group communications using key graphs. IEEE/ACM Transactions on Networking. 2000; 8(1):16–30.
25. Wallner D, Harder E, Agee R. Key management for multicast: Issues and architectures. RFC 2627; 1999.
26. Adrian P, Song D, Tygar JD. ELK, a new protocol for efficient large-group key distribution. IEEE Proceedings of Symposium on Security and Privacy, S&P 2001; 2001. p. 247–62.
27. Alan TS, Mc Grew D. Key establishment in large dynamic groups using one-way function trees. IEEE Transactions on Software Engineering. 2003; 29(5):444–58.
28. Ballardie A. Scalable multicast key distribution. RFC 1949; 1996.
29. Ballardie A. Core Based Trees (CBT version 2) multicast routing protocol specification. RFC 2189; 1997.
30. Suvo M. Iolus: A framework for scalable secure multicasting. ACM SIGCOMM Computer Communication Review. 1997; 27(4):277–88.
31. Dondeti LR, Mukherjee S, Samal A. A dual encryption protocol for scalable secure multicasting. IEEE Proceedings of International Symposium of Computers and Communications; 1999. p. 2–8.
32. Du F, Ni LM, Esfahanian AH. Towards solving multicast key management problem. Proceedings of 8th IEEE International Conference on Computer Communications and Networks; 1999. p. 232–6.
33. Briscoe B. MARKS: Zero side effect multicast key management using arbitrarily revealed key sequences. Networked Group Communication, Springer Berlin Heidelberg; 1999. p. 301–20.
34. Molva R, Pannetrat A. Scalable multicast security with dynamic recipient groups. ACM Transactions on Information and System Security (TISSEC). 2000; 3(3):136–60.
35. Setia S, Koussih S, Jajodia S, Harder E. Kronos: A scalable group re-keying approach for secure multicast. Proceedings IEEE Symposium on Security and Privacy, S&P; 2000. p. 215–28.
36. Hardjono T, Cain B, Monga I. Intra-domain grouop key management protocol. IETF Draft; 2000.
37. Rafaeli S, Hutchison D. Hydra: A decentralised group key management. Proceedings of 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises; 2002. p. 62–7.
38. Marsson I, Tang DT, Wong CK. A conference key distribution system. IEEE Transactions on Information Theory. 1982; 28(5):714–20.
39. Steiner M, Tsudik G, Waidner M. Diffie-Hellman key distribution extended to group communication. Proceedings of 3rd ACM Conference on Computer and Communications Security; 1996. p. 31–7.
40. Boyd C. On key agreement and conference key agreement. Information Security and Privacy. Springer Berlin Heidelberg; 1997. p. 294–302.
41. Becker K, Wille U. Communication complexity of group key distribution. Proceedings of the 5th ACM Conference on Computer and Communications Security. 1998. p. 1–6.
42. Dondeti L, Mukherjee S, Samal A. A distributed group key management scheme for secure many-to-many communication. Department of CS, University of Maryland, Tech Rep, PINTL-TR-207-99; 1999.
43. Rodeh O, Birman K, Dolev D. Optimized group rekey for group communications systems. Cornell University; 1999.
44. Perrig A. Efficient collaborative key management protocols for secure autonomous group communication. International

Workshop on Cryptographic Techniques and E-Commerce (CrypTEC'99); 1999. p. 192–202.
45. Kim Y, Perrig A, Tsudik G. Simple and fault-tolerant key agreement for dynamic collaborative groups. Proceedings of the 7th ACM Conference on Computer and Communications Security; 2000. p. 235–44.
46. Steer DG, Strawczynski L, Diffie W, Wiener M. A secure audio teleconference system. Proceedings on Advances in Cryptology. Springer-Verlag; 1990. p. 520–8.
47. Kim Y, Perrig A, Tsudik G. Communication-efficient group key agreement. Trusted Information. US: Springer; 2001. p. 229–44.
48. Kim Y, Perrig A, Tsudik G. Tree-based group key agreement. ACM Transactions on Information and System Security (TISSEC). 2004; 7(1):60–96.
49. Rao GSVRK, Radhamani G. WiMAX: A wireless technology revolution. Auerbach Publications; 2008.
50. Ahson SA, Ilyas M. WiMAX Applications. CRC press; 2007.
51. Glisic SG. Advanced Wireless Networks: 4G Technologies. John Wiley Inc; 2006.