BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 3

Sofia • 2015

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0038

A Survey on Group Key Management Schemes

R. Seetha, R. Saravanan

VIT University, Vellore, Tamilnadu, India
Emails: rseetha@vit.ac.in rsaravanan@vit.ac.in

Abstract: *Cryptographic key management needs utmost security in generating, exchanging, using, rekeying, storing of keys, being used for communication purposes. Successful key management is critical to the security of a cryptosystem. This paper presents a detailed survey on group key management and its challenges in network independent and network dependent approaches. The paper also focuses on the advantages, disadvantages and security vulnerabilities of these protocols.*

Keywords: *Group key, security, rekeying, multicast, key management.*

1. Introduction

The importance of group communication which involves more than two nodes can be well understood from the on-going real time applications, such as email, Skype, chat, Facebook, Twitter and online games, etc. Though group communication has found rapid growth in today's networking environment, security remains a great challenge. Apart from the social networks, more secured environments like a military network, where many sensitive data are being exchanged, require confidential and secure environment for data transmission, membership management and key management. Thus, the security of group communication depends on the secrecy and strength of the group key used.

Group key establishment can be key agreement and key distribution. Key agreement requires each participating node contribution to generate a key and assure that it is newly generated. In a key distribution scheme one participating node is responsible for generating and distributing the key to all participating nodes of the group communication. Another important feature is a rekeying process when the membership changes in the dynamic environment.

The group key management protocols can be generally classified (Fig. 1) as Network independent based and Network dependent based key management

protocols [53]. The network independent based key management protocol is further classified into centralized, decentralized and distributed key management protocols, whereas the network dependent based key management is classified as tree based and cluster based key management.

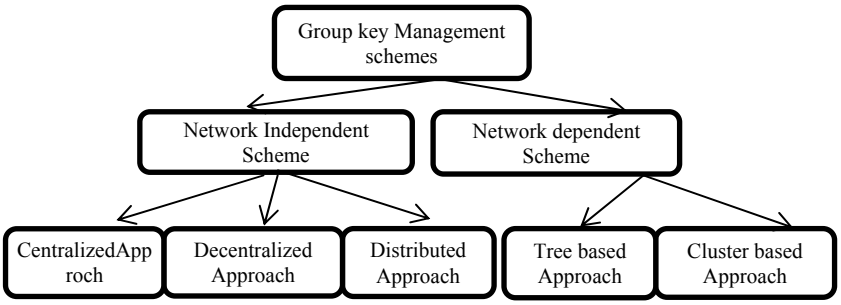

Fig. 1. General classification of group key management schemes

The role of key management also extends to a participating member authentication, to prevent any intruder from impersonating and providing access control to validate the joining operation. Moreover, the key management follows a set of cryptographic techniques in generating and distributing the keys which may be symmetric or asymmetric for secure group communication.

2. Requirements of group key management

The basic requirements of security are confidentiality, integrity and authenticity. The major requirements can be grouped as performance requirements, efficient functioning requirements and security requirements [2-4].

2.1. Performance requirements

The factors that affect group communication include scalability (ability to handle the dynamic group size), 1-affects-*n* phenomenon (carrying out a rekeying process when a member joins or leaves the group), delays, a single point failure, availability of resources, such as bandwidth for efficient communication.

2.2. Efficient functioning or qos (**q**uality **o**f **s**ervice) requirements

For the efficient functioning of key management protocols, care should be taken of the control packets overhead, storing of keys and the amount of time taken to encrypt and decrypt the keys to be used.

2.3. Security requirements [1]

- Denying the left member from accessing future keys of the group – *Forward secrecy*.
- Denying the new member from accessing previous keys of the group – *Backward secrecy*.

- The generated keys should be completely independent from the previous ones to avoid key prediction – *Key independence*.
- Deploying security analysis techniques and a threat model to prevent and protect the keys from both inside and outside attack of the group – *Resilient*.

3. Group key management schemes

Wireless network, as shared medium is an environment, where any user in the network can access the packets delivered. Access to the group can be enforced through encryption techniques. Thus, a shared key can be used to encrypt the communication to prevent unauthorized users from accessing the communication. The shared encryption key is called a group key or a Traffic Encryption Key (TEK). It is the key, on which the security of group communication is entirely dependent.

3.1. Centralized key management scheme

In a centralized system, a single entity is responsible to carry out group communication. Key generation, distribution and management are all carried over by this entity. The major challenges of a centralized scheme include:

- *Scalability overhead*: As the success of group communication depends on the single centralized entity, the task of rekeying becomes an overhead when the group size increases.
- *Storage overhead*: The number of keys to be secured for a session.
- *Single point of failure*.
- *Maintaining forward and backward secrecy*: A new member joins or an old member leaves the group.
- *Collusion independence*: Co-operation among expelled members to work together and share their own piece of information to regain access to the group key.

The centralized key management scheme can be further categorized into:

- **Pairwise key approach.** In this approach, the single point entity shares pairwise keys with each participating member of the group. An example protocol of this approach is Group Key Management Protocol (GKMP). Harney and Muckenhirn [6, 7] proposed GKMP which combines creation of pairwise keys with techniques used to distribute keys from a Key Distribution Centre (KDC) to distribute a symmetric key to a multicast group member.
- **Group Key Centre identity.** The initiator of the multicast group obtains a group management certificate from its certification hierarchy. The certificate holder becomes the sole responsible for generation and distribution of the group key.
- **Creating a Group Key.** On behalf of the certificate holder, the Group Key Management (GKM) application selects a group member and creates a Group Key Packet (GKP) on contacting it. GKP holds the pairwise key (the current Group Traffic Encrypting Key (GTEK) and a future Group Key Encrypting Key (GKEK)):

$$GKP = [GTEK_n, GKEK_{n+1}].$$

- **Distributing Group Key.** The GKM contacts each member of the group, validates and creates a group session key (session TEK and KEK) and group rekey (E_{KEK} (GKP)), signed using the originator's certificate.
- **Rekeying.** When a new member joins the group, the GKM application acting as an originator, creates a new GKP and a new group rekey (encrypted using GKEK) and broadcasts to the group members.
- **Secure Lock approach.** Chiou and Chen [5] proposed a centralized key management approach where the Single point entity establishes a group key or a rekey process when a member leaves in a single broadcast. The central entity performs Chinese Remainder computation on each message before sending it to the group member, but the number of rekey messages is considerably reduced. Here in this approach, the central entity allocates a positive integer m_i and shares a secret value k_i with each group member. Whenever the central entity wants to send a message to the group member, it generates a random value K with which the message is encrypted. The value K is in turn encrypted with each secret k_i where $i = 1,\ldots, n$. Then a lock M is computed as follows:

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

The lock M and the message encrypted with K is sent to the group members. On receiving the lock M, each member recovers the encryption key K and decrypts the message received. Only members whose secret k_i and its corresponding positive integers m_i are included in the computation of the lock M, can alone recover the key K.

- **Hierarchy of Keys Approach.** In this approach the central entity reduces the overhead caused by rekeying, by sharing the secret keys with subgroups of the entire secure group. Thus, when a member leaves the current session, the central entity uses the secret keys shared with subgroups to distribute the new TEK since the shared secret keys are unknown to the leaving member. This approach also uses pairwise keys and trades off storage overhead, as the overhead of rekeying is reduced. The protocols following the hierarchy approach, are Logical Key Hierarchy (LKH), One way Function Trees (OFT) and Centralized Flat table Key Management (CFKM).

o *Logical Key Hierarchy*: In LKH (Wallner, Harder and Agee [8], Wong, Gouda and Lam [9, 10]) the root of the tree holds the TEK. The nodes at interior levels of the tree holds keys along the path from a leaf to itself and are named as KEKs and the leaves of the tree holds the secret keys shared with the group members. For a balanced binary tree, each member stores at most $1 + \log_2 n$ keys, where n is the number of group members.

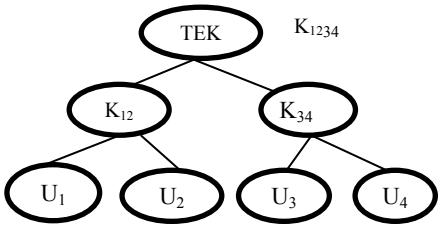

Fig. 2. Key hierarchy

Fig. 2 explains the scenario of a multicast session, using LKH protocol among four members of the group: U_1, U_2, U_3, U_4 are the secret keys of each group member; K_{12} is the group key of members 1 and 2; K_{34} is the group key of members 3 and 4; TEK is the key, comprising the key shared among the group. Assume that U_2 wants to leave the group. The root updates U_1 with the message K_{12}' and sends it to U_1 encrypted, using K_1. Thus the number of messages required for the rekeying process is reduced compared to GKMP. LKH approach is also extended to k-ary key trees and it is observed that as the degree increases, the number of keys to be maintained by the group members is reduced because of a smaller tree depth.

One way Function Trees (OFT). OFT is an extension work of LKH. The number of rekey messages is reduced to $\log_2 n$. In OFT, proposed by D. Balenson, D. McGrew and A. Sherman [11, 13], KEKs are computed by the group members, whereas in LKH it is carried out by the root entity. For each node, KEK (K_i) is calculated using its left and right child KEKs and one way function g given as:

(1) $$K_i = f(g(k_{\text{left}(i)}), g(k_{\text{right}(i)})).$$

The $g(k)$ computed is called a blinded key and f is a mixing function (XOR operation). Each member of the group holds its secret key at the leaf node, its sibling's blinded key and a set of blinded sibling keys of its ancestor node, where the ancestors of a node are the nodes in the path from its parent node to the root.

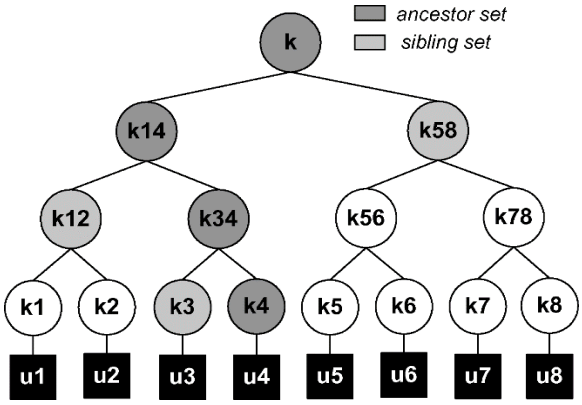

Fig. 3. Ancestor and sibling key sets of U_4

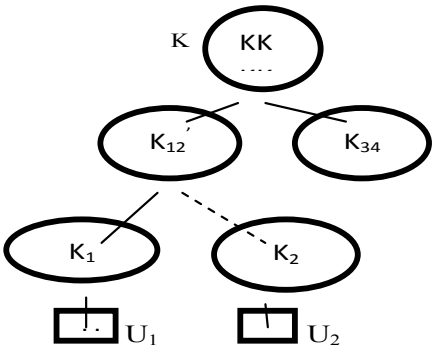

Fig. 4. Member U_2 joining the group

The scenario of OFT can be explained using Fig. 4. Assume that a member having the secret key U_2 wants to join the group. When the user U_2 joins the group, the keys K_1 and the group key K should be modified into K_{12}', K. For updating the new group key and the modified KEK (K_{12}'), the only value needed to be sent is the blinded key $g(K_2)$, $g(K_{12}')$, encrypted by using the sibling blinded key sets $g(K_1)$, $g(K_{34})$ and the new TEK and KEKs are calculated using formula (1) in a recursive manner. For example, $K_{12}' = f(g(K_1), g(K_2))$. Thus in this approach the number of messages required for rekeying is reduced to half the number of messages in comparison to LKH approach. Similar to OFT, one other approach exists, called one way Function Chain tree (Canetti et al. [12]). The performance of this approach is similar to OFT, but generates new KEKs, using a pseudo random generator instead of using one way function.

o *Centralized Flat Table Key Management*. To reduce the overhead of maintaining keys by the central entity, the CFKM approach proposed in Waldvogel et al. [14] uses flat tables to store the keys. The table entries include one TEK and $2w$ more entries for KEKs, where w is the number of bits in the member id (preferably IP address). Table 1 shows the keys associated with each bit of member ID. Thus, each member is assigned w+1 keys, where w is the number of KEKs and one TEK. As an example for a member with ID 1010 holds $KEK_{0,1}$, $KEK_{1,0}$, $KEK_{2,1}$, $KEK_{3,0}$ and TEK.

Table 1. Flat table for $w = 4$

TEK	
$KEK_{0,0}$	$KEK_{0,1}$
$KEK_{1,0}$	$KEK_{1,1}$
$KEK_{2,0}$	$KEK_{2,1}$
$KEK_{3,0}$	$KEK_{3,1}$

Table 2. Rekey process after ID 1010 leaves

TEK	
$(TEK_{new})KEK_{0,0}$	$(KEK_{0,1new})(TEK_{new})$
$(KEK_{1,0new})(TEK_{new})$	$(TEK_{new})KEK_{1,1}$
$(TEK_{new})KEK_{2,0}$	$(KEK_{2,1new})(TEK_{new})$
$(KEK_{3,0new})(TEK_{new})$	$(TEK_{new})KEK_{3,1}$

Assume that a member leaves the group. The rekeying process is initiated by the central entity, sending a re-key message. The message has two parts: the first part is TEK encrypted with KEK that is not changed which allows all the participating members to decrypt the new TEK. The second part has new KEKs encrypted, using the old KEK and new TEK, thus providing forward secrecy. As an example, Table 2 figures out the rekeying process when a member with ID 1010 leaves the group.

- **Summary.** Table 3 summarizes the performance of the Centralized key Management protocols discussed so far. The table shows how factors, such as

1-affects-*n*, forward secrecy, backward secrecy, collision freedom, storage overhead, rekey overhead have their impact on the CKMPs.

Table 3. Comparison of Centralized key management schemes (*K*-key size in bits, *d*-height of the tree, *I*-number of bits in member ID, *n*-number of the group members)

Protocol	1-affects-*n*	Forward secrecy	Backward secrecy	Collusion Freedom	Rekey (Multicast)		Storage overhead		Remarks
					Join	Leave	KDC	Member	
GKMP	Yes	No	Yes	Yes	$2K$	–	$2K$	$2K$	Rekey overhead
LKH	Yes	Yes	Yes	Yes	$(2d–1)K$	$I+2dK$	$(2n–1)K$	$(d+1)K$	Storage overhead
OFT	Yes	Yes	Yes	Yes	$(d+1)K$	$I+(d+1)K$	$(2n–1)K$	$(d+1)K$	Storage overhead
CFKM	Yes	Yes	Yes	No	$2IK$	$2IK$	$(2I+1)K$	$(I+1)K$	Storage overhead
Secure Lock	No	No	No	Yes	–	–	$2nK$	$2K$	Computation overhead

3.2. Decentralized group key management

The key idea of Decentralized Group Key Management scheme is to reduce the load on KDC, the central entity. This is achieved by splitting the group members into several subgroups and each subgroup is managed by its own subgroup controller. This approach solves the problem of a single point failure. The decentralized approach can be further categorized as membership driven protocols (the rekeying process carried out as the member leaves or joins the group) and time driven protocols (rekeying is carried out at certain time intervals). Protocols like Scalable Multicast Key Distribution (SMKD), Intra-domain Group Key Management Protocol (IGKMP), Hydra fall under the category of membership driven. Kronos, MARKS, Dual-Encryption Protocol (DEP) are examples of time driven category. The challenges in the decentralized scheme: The first one is how efficiently this scheme collaborates with other group key management schemes to distribute key messages to subgroup members. The second one is establishing the trust relationship among the third parties which are involved in decentralized schemes. The third one is authenticating the members of a subgroup participating in the session which may be in the same or different networks.

- **Scalable Multicast Key Distribution.** SMKD proposed in Ballardie [15, 16], Ballardie, Francis and Crowcroft [17] uses a Core Based Tree (CBT) multicast routing protocol for constructing a multicast tree. The CBT has a main central entity, as well as secondary entity cores. The main core entity creates an Access Control List (ACL), a group session key (GTEK) and a key encryption key (GKEK) for updating GTEK. These keys are transmitted to secondary cores and other nodes when they join the multicast tree after authenticating the joining nodes. The main core entity authenticates the secondary core which in turn authenticates the joining members and uses the ACL to distribute the keys, but the keys for the session are generated only by a central core. In SKMD, the problem of forward secrecy remains unsolved.

- **Intra-domain group key management.** The IGKMP architecture proposed by DeCleene et al. [18, 19] divides the network into administratively scoped areas, such as Domain Key Distributor (DKD) and Area Key Distributor (AKD) for each available area. The DKD is responsible for generating the group key TEK and shares with the group members through AKD.

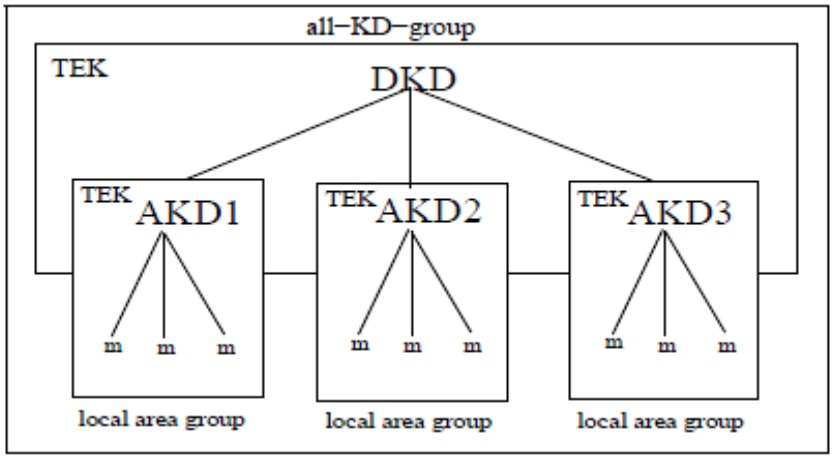

Fig. 5. IGKMP architecture

Fig. 5 shows IGKMP architecture. The DKD and AKD are placed in a multicast group, named all-KD-group which is used by the DKD to transmit the rekey messages to all AKD. All areas in the domain use the same group key. DKD keeps a track of AKD which in turn tracks the members of the group. The failure of DKD, the central controller leads to a failure in the entire group communication and failure of AKD leads to interruption of the communication of that area, since it is the only central point of communication for that particular area.

- *Hydra*. Hydra protocol proposed by Rafaeli and Hutchison [20] is a decentralized group key management scheme where the session group is divided into smaller subgroups. These subgroups are controlled by a server called Hydra Server (HS). The group key is distributed to all HS, using a Synchronized Group key Distribution Protocol (SGKDP) and it is ensured that a single trusted HS is generating the new group key whenever a leaving or joining operation occurs. The failure of one or more HS does not interrupt the multicast session.
- *Kronos*. It is a time driven approach, in which rekeying takes place at periodic intervals of time, irrespective of members joining or leaving the group. Kronos, proposed by Setia et al. [21] uses architecture similar to IGKMP. Unlike IGKMP, AKD is responsible for generating the group key and relaying it to the members of its area at periodic intervals of time. In order to have the same group key relayed after some period of time, the clocks of all AKDs are synchronized so that all AKDs agree upon with the rekeying time period. For clock synchronization it is preferred to use the Network Time Protocol (NTP). In addition to synchronization, all AKDs must agree upon two secret factors, namely R_0 (an initial value) and K, the master key which are sent by DKD over a secure channel. These secret factors are used by AKDs to generate future keys, say R_{i+1} where $i > 0$; R_1 is

obtained by encrypting R_0, using K, and $R_1=E_K(R_0)$, and thus future keys $R_{i+1}=E_K(R_i)$. Kronos overcomes the single point failure problem as rekeying is carried out independently by AKDs, but security of the group remains open as rekeying depends on the previous group key.

- *MARKS*. Briscoe [22] proposed MARKS. In this time driven approach the time slices, at which rekeying is carried out, is protected by encrypting each time slice with different keys. The architecture uses a binary hash tree in which the encryption keys form the leaf nodes which are generated from a single seed. The interior nodes are also called seeds. The tree is constructed by applying a blinding function MD_5 on the seeds as follows:
- The depth D of the tree is determined to define the number of keys (N) required, $N=2^D$.
- The seeds of the tree are represented as $S_{i,j}$, where i indicates the depth of the tree and j indicates the key number at depth i. Choosing randomly, the root seed is $S_{0,0.}$
- Using the parent seed, the left and right seeds are generated. The **l**eft **s**eed (ls) is generated by shifting the parent seed one bit to the left and applying MD_5 on the shifted bits. The **r**ight **s**eed (rs) is generated by shifting the parent seed one bit to the right and applying MD5 on the shifted bits:

$$S_{1,0}= MD_5(ls(S_{0,0}),$$
$$S_{1,1}= MD_5(rs(S_{0,0}).$$

- Similarly the other intermediate seeds are generated up to the decided depth D.

The users participating in the communication generate the keys on receiving the seed. This approach cannot do a rekeying process whenever there is a change in the group membership, since the keys are renewed after a certain time slice.

- **Dual-Encryption Protocol (DEP).** This decentralized time driven protocol solves the problem of trusting third parties, because many intermediate nodes are available in the decentralized approach. DEP proposed by Dondeti, Mukherjee and Samal [23, 24] subgroups the members of the group in a hierarchical manner and the subgroups are controlled by a Sub-Group Manager (SGM). Here three KEKs and one DEK (Date Encryption Key) is used. KEK_{i1} is shared between a SGM_i and its subgroup members. KEK_{i2} is shared between the Group Controller (GC) and the members of subgroup i, excluding SGM_i. Finally, KEK_{i3} is shared with SGM_i by GC. DEK, generated by GC is communicated to the group members, the GC by encrypting it with KEK_{i2} and encrypted again with KEK_{i3}. On receiving the encrypted packet of DEK, SGM_i decrypts, using KEK_{i3} and again encrypts the encrypted DEK, using KEK_{i1} that is shared among the subgroup members and sends it to subgroup i. Now, each member of subgroup i decrypts the message first using KEK_{i1} and then decrypts, using KEK_{i2} and recovers DEK. DEK could be obtained by the members only, which know both the keys. Thus SGMs becomes a trusted third party, since the access of DEK is not possible for them as they do not know KEK_{i2}. Whenever there is joining or leaving of a member in subgroup i, SGM_i changes KEK_{i1} and sends it to its members currently participating. The changes in DEK will not allow access to the members of

subgroup i, which have not received the new KEK_{i1}. If DEK remains the same, still the members which have not received KEK_{i1} can still access the multicast session thus leading to a forward secrecy challenge.

- **Summary.** The performance of the membership driven approach and the time driven approach protocols of a decentralized group key management scheme are summarized in Table 4. The protocols are compared, using parameters like key independence, rekeying among subgroup members and central rekeying. In addition to these parameters, the protocols are remarked as fault tolerant or not.

Table 4, Summary of decentralized key management schemes

Protocol	Key independence	1-affects-*n*	Local rekey	Rekey	Fault tolerant
SKMD	Yes	Yes	No	No	No
IGKMP	Yes	Yes	No	Yes	No
Hydra	Yes	Yes	No	Yes	Yes
Kronos	No	No	No	No	Yes
MARKS	No	No	No	No	Yes
DEP	Yes	Yes	No	No	No

3.3. Distributed group key management

In the distributed GKM approach the group members of the multicast session cooperate with each other to generate the required group key. In this approach there is no group controller and this makes the system fault tolerant. On the other hand, the distributed key management scheme compromises the security mechanisms, when there is a change in the group membership; secondly, the processing time and communication overhead increase when the group size increases; thirdly, each member has to keep a track of the other members participating in the multicast session to make robust communication. This key management approach is further categorized as Ring based cooperation; Hierarchy based cooperation and Broad cast based cooperation, based on the virtual topology created by the cooperating group members. The parameters affecting the distributed approach are:

- Number of the round required for processing and communicating.
- Number of the messages to be exchanged among the group members.
- Computational cot to generate the group key.

3.3.1. Ring based cooperation

In ring based group key management, the participating members form a virtual ring.

- **Ingemarson et al. protocol**, proposed by Ingemarson et al. [25] is an example protocol which is based on this category. This protocol is an extension work of Diffie Hellman key agreement protocol for group communication. Here:
  - The group members are organized into a virtual ring; such that the member M_i, $i = 1, \ldots, n$, communicates with the member M_{i+1} and the last member M_n with the member M_1.
  - The group key is computed within $n-1$ rounds.

o Each member M_i generates N_i randomly and computes g^{N_i} and sends it to the next member M_{i+1}.

o Similarly, each member M_i performs n exponentiations and gets the group key $K_n = g^{N_1 N_2 \ldots N_n}$ after $n-1$ rounds.

Whenever there is a change in the group membership, the entire algorithm has to be repeated to generate the new group key.

- **Diffie Hellman for Multicast (DFM)**. This protocol work of Steiner, Tsudik and Waidner [26] is an extension of two parties, Diffie Hellman key exchange protocol [27] to n party communication. Each member of the group agrees with two primes α and q. Each member chooses its own secret key, say x_1, x_2, …, x_n. Initially, the first member computes α^{x_1} and sends tit o the second member which computes $\alpha^{x_1 x_2}$. Similarly, the remaining members of the group raise their own secret key to the received intermediate values and thus the last member of the group easily computes the final group key $k = \alpha^{x_1 \ldots x_n} \bmod q$. Upon computing the final key, the last member multicasts the key to the entire group. Each member, on receiving the group key extracts their respective intermediate value and generates k. As the number of participating members increases, the length of the message increases and so the operation of exponentiation.

3.3.2. Hierarchy based cooperation

- **Octopus.** Becker and Wille proposed Octopus protocol [28]. This protocol is also an extension work of Diffie Hellman Key (DHK) exchange. Here, the multicast group is divided into subgroups, each subgroup containing four members. Each subgroup agrees upon and computes the intermediate key I_{subgroup} value and exchanges it with the other subgroups. The leader of each subgroup is responsible for exchanging the intermediate key I. Assume that there are four subgroups with leaders A, B, C, D. First A and B exchange their intermediate keys I_A and I_B and compute $\alpha^{I_A \cdot I_B}$. Similarly, C and D compute $\alpha^{I_C \cdot I_D}$. Now A and C exchange $\alpha^{I_A \cdot I_B}$, $\alpha^{I_C \cdot I_D}$ and compute the $\alpha^{I_A \cdot I_B \cdot I_C \cdot I_D}$. Similarly, B and D compute and thus all subgroups are capable of computing the final required group key.

- **Skinny Tree Protocol (STR).** The protocol STR, proposed by Kim, Perrig and Tsudik [29] uses a tree structure. Fig. 6 below shows an example of STR tree with four members. The members of the group are organized as leaf nodes (LN_i, where $i = 1, ..., n$). Each leaf (group member) holds a secret random value r_i and computes its public blinded key $br_i = g^{r_i} \bmod p$ (g and p are diffie Hellman parameters). The Internal Node (IN) is identified as IN_i in the tree and holds a secret random k_i and its blinded public key $bk_i = g^{k_i} \bmod p$. Similarly, each node secret k_i is recursively calculated as follows: $k_i = (bk_{i-1})^{r_i} \bmod p = (br_i)^{k_{i-1}} \bmod p$. The root node holds the group key. The tree is organized in a linear manner and hence, this protocol takes $O(n)$ time to establish the group key. Each member of the tree is required to store and maintain all the public keys associated to all the nodes of the tree. When a member joins or leaves, the tree is re-built and all the members update the group key to construct a new key k_n associated to the root of the tree.

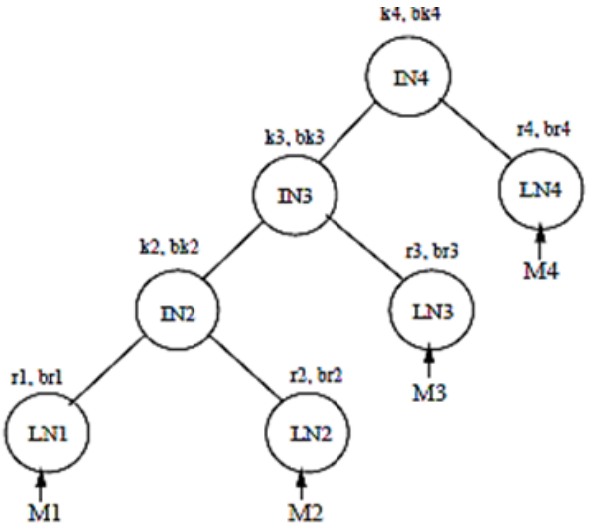

Fig. 6. STR tree with four members

- **Distributed Logical Key Hierarchy (D-LKH).** Unlike the centralized LKH protocol, D-LKH work of R o d e h, B i r m a n and D o l e v [30] has no GC. The hierarchy generated is grouped into a left subtree and right subtree with two subtree members agreeing on a mutual group key for encryption. The member m_{left} is assumed to be the left subtree leader and the member m_{right} is assumed to be the right subtree leader. The key for the left subtree L and the right subtree R is KL and KR, respectively. Mutual key agreement between the left and right subtree is as follows:
  - The member m_{left} chooses a new key K_{LR}, and sends it to member m_{right} through a secure channel.
  - Member m_{left} encrypts the key K_{LR} with key KL and distributes it to the members of subtree L; similarly m_{right} encrypts the key K_{LR} with key KR and distributes it to members of the subtree R.
  - Now all members of the entire tree have received the new group key.

  The above algorithm takes three rounds and each member stores three keys. The total number of rounds taken to complete is $\log_2 n$ rounds with each member storing $\log_2 n$ keys.
- **Distributed One-way Function Tree (D-OFT).** D-OFT proposed by D o n d e t i, M u k h e r j e e and S a m a l [31] is similar to the centralized approach, but the difference is that this protocol has no group controller GC. Every group member is trusted with the access control and key generation. Similar to the centralized approach, every member generates its own key and sends its blinded version of the key to its sibling. Thus every member knows all the keys along the path from its node to the root node and all the blinded keys from the sibling node to the nodes along the path to the root.
- **Distributed Flat Table (D-FT).** In this scheme, proposed by W a l d v o g e l et al. [14], each member knows only their KEKs. The inconvenient

part of the distributed scheme is that a newly joining member is to contact a group of members to get all the keys needed. This may lead to a delay in key synchronization since many members may be involved in changing the same key at the same time.

3.3.3. Broadcast based cooperation

- **Fiat and Naor protocol.** This protocol is based on Diffie Hellman property. In this protocol, proposed by Fiat and Naor [32], a trusted reliable centre T initializes the system. T chooses two primes p and q and broadcasts $n = p.q$ to all nodes. Then T generates a random number g and keeps it secret. When a new member M_i joins the group, T sends to this new member two values:
  - a random x_i (which is relatively prime with each other x_j previously generated for the members M_j);
  - a key $\alpha_i = g^{x_i} \bmod n$. M_i keeps α_i secret.

  To agree upon a group key K, each member broadcasts its values x_i and hence, each of them calculates K.
- **Burmester and Desmedt protocol (BD).** Burmester and Desmedt [34] proposed an efficient distributed key management protocol that requires only three rounds. The algorithm is as follows:
  - the member m_i generates its random exponent r_i and broadcasts $Z_i = \alpha_{r_i}$;
  - the member m_i computes and broadcasts $X_i = (Z_{i+1}/Z_{i-1})r_i$;
  - the member m_i computes the key K_n, which is given as $K_n = \alpha N_1N_2+N_2N_3+...+N_nN_1$.

  This protocol requires n+1 exponentiations per member and in all but one the exponent is at most $n-1$. The drawback of this protocol is that it requires $2n$ broadcast messages.
- **Conference Key Agreement (CKA).** CKA proposed by Boyd [33], a distributed approach where all group members take part to generate the group key. The group key is generated using a combining function $K = f(N_1, h(N_2), ..., h(N_n))$, where f is the combining function (a MAC), h is a one-way function, n is the group size, and N_i is the contribution from a group member i. The protocol specifies that $n-1$ members broadcast their contributions (N_i). The group leader, for example U_1, encrypts its contribution (N_1) with the public key available with each member and broadcasts it. All group members who have the public key can decrypt and generate the group key.
- **Summary.** Table 5 summarizes the requirements of the Distributed key exchange schemes. The comparison is done in terms of number of rounds, number of multicast messages required, Diffie hellman keys and leaders required. This scheme overcomes the problem of 1-affects-n problem because in the distributed scheme all group members take part in generating the required group key and thus every member must again perform all required computations when a member joins or leaves the group.

Table 5. Summary of Distributed Key Management schemes

Protocol	Rounds	Multicast messages	DH Key	Leader requirement
Ingemarson et al	n–1	–	Yes	No
DFM	N	n	Yes	No
Octopus	2(n–1)/4+2	–	Yes	Yes
STR	N	n	Yes	No
D-LKH	3	1	No	Yes
D-OFT	$\log_2 n$	–	No	No
D-FT	N	–	No	Yes
Fiat-Naor	2	n	Yes	Yes
BD	3	2n	No	No
CKA	3	n	No	Yes

4. Network dependent schemes

Group key management protocols that are network independent cannot be applied for wireless networks. For performing an efficient group key management over wireless networks, group key management protocols should be network dependent. To efficiently deploy or operate such group key management protocols, they must depend on the features of the basic network infrastructure. The major challenges of the Network dependent group key management protocols is supporting mobile multicast where members move over the wireless network and continue receiving their subscribed multicast services. The movement of members over the network needs to be handed off from one area to another area in order to get multicast services efficiently and thus induce more complexity on handling key management and traffic control. The challenge of key handling, whenever a member leaves or joins the group increases the complexity both on intra and inter domain of the group, such that the security of the data transmitted is preserved and the overall system performance also remains unaffected. The mobile multicast security group key management protocols listed are classified into tree based and cluster based approaches.

4.1. Tree based approach

- **Topology Matching Key Management Tree (TMKM).** Yan, Trappe and Liu [35, 36] proposed TMKM. This protocol uses a LKH key tree and matches the key management tree to a three-level topological structure. The structure uses three network entities, namely the mobile users, Base Stations (BS) and a Supervisor Host (SH). The BSs controlled by SH perform key management within its cell and multicasts the information about the group key to its members. The SH takes care of mobile users routing and generates the required key, including the group key (TEK) and the supporting keys for secure group communication. When the members move between cells, an efficient handoff mechanism handles the relocation of that user in TMKM tree. Each cell is associated with a Wait to be removed list (WTB) that tracks previous and current cell members. TMKM

protocol has low communication overheads by broadcasting rekey messages to only useful members of the cell. But the use of common TEK leads to 1-affect-n problem and thus the rekeying process affects the entire multicast session. BSs and SHs are provided by a third party. TMKM uses centralized structure, thus causing a single point of failure. The physical position of a member on the tree is more important since the members moved need to be relocated on the TMKM tree and TMKM is incapable of handling frequent handoffs.

- **A Hybrid Key Management Scheme (HKM).** HKM proposed by Lin, Xueming and Yong [37] for wireless environment is similar to TMKM and has Topology Matching (TM) sub trees and Topology Independent (TI) sub-trees. HKM tree uses two key management trees, namely, TIKM trees and TMKM trees and combines the features of these trees to manage high mobility and low mobility members, thus minimizing the rekeying overheads that occur during the handoff process. These schemes enable the rekeying messages for low mobility users to be delivered to the specified location and the rekeying messages for high mobility users only need to be broadcast when the users leave the group, regardless of the number of handoffs occurring. The HKM tree is partitioned into two sub-trees: the TI sub tree is for high mobility users and the TM sub tree is for low mobility users. The high mobility and low mobility is measured based on the velocity, by which the member moves. If V is the velocity of members moving and if $V > V_0$ then the members are called high mobility members; otherwise they are called low mobility members, where V_0 is the threshold velocity. Every individual member of the group holds the private key, the session key and a set of KEKs on the path from itself to the root node of the key tree. HKM performs two operations; a rekeying process for high mobility users when the member joins or leaves the group and locating the delivery of the rekeying messages to low mobility members. Unlike TKMM, HKM combines features of TIKM trees and TMKM trees to handle both low and high mobile members, such that their physical location does not affect the performance of key management and the bandwidth is efficiently used for members which do not move frequently. But the bandwidth is highly used for high mobility users, as the rekeying messages are required to be broadcast to all the BSs. Like TKMM, HKM, uses the common TEK approach, thus it suffers from 1-affect-*n* phenomenon.
- **WANG Approach.** This approach, proposed by Yiling, Phu Dung, and Srinivasan [38, 39] is a distributed network dependent group key management protocol where the group members are divided into leader units and general member units. The leader units handle the key management. This protocol also takes care of the null area re-keying when the members move between two cells. This approach proposes a handoff member mechanism to handle the member mobility. The mechanism consists of 2-tier logical framework to match the cellular network topology. This 2-tier framework has a key server which significantly reduces the communication overhead that occurs during the key updating. The network entities used are: Group Key Server (GKS) and Independent Cell Key Servers (CKS). GKS occupies the first level, generates a group key and multicasts the keys to control the units at the second level. The second level is occupied by CKS. CKS, allotted to each cell receives the group key from the GKS and

redistributes it to the members of the cell. Logically, CKS partitions its group members into two roles namely, the leader unit at the leader level and the general member unit at the user level. Each member unit is allotted a leader. The leader helps CKS in key distribution. The participating group members at the user level take care of the rekey process when the membership changes. This rekey process is known as micro key management. This approach follows a decentralized framework for scalability and helps to reduce the rekeying overhead. In this approach the key management operations are performed in parallel in all the cells simultaneously. The other side of WANG approach fails to preserve backward and forward secrecy. Storage of a large number of keys by each mobile member affects the operational performance of the members of multicast session in the wireless network. Authentication delay is caused when a huge number of members starts moving since they need to contact the old (previous) area server each time when they move to a new area. In addition, the increased number of levels complicates the key management and delays the packet delivery.

- **Combination of Rekeying and Authentication in Wireless networks (CRAW).** This protocol combines the member authentication mechanism with the group key management for efficient rekeying. For membership authentication Simple And Secure (SAS) password authentication protocol (Sandirigama, Akihiro and Noda [41]) is used and Code for Key Calculation (CKC) (Hajyvahabzadeh et al. [42]) protocol is used for group key management over a dynamic topological network. The combination provides a simple and secure mechanism for mobile members joining or leaving a group or moving between cells. This protocol follows a decentralized framework in a cellular wireless network. The protocol has a main server which distributes the multicast details to an individual Area Wireless Server (AWS) and maintains a list, containing the member information regarding joining and leaving the group and movement between cells. AWS is responsible for performing the member authentication process, generating and sending the Area group key, and it also forwards the multicast details to the mobile members. CRAW [40] uses an efficient network independent key management protocol CKC (an improved version of LKH) to manage the rekeying process in each subgroup. Whenever there is a change in the group membership, the group members compute the necessary u-node and k-node keys of the tree, using node codes and one-way hash function on receiving the updated group key after the membership change. This reduces the workload on the server considerably. The use of the one way hash function helps in keeping the key secure. The operation has two phases: one is the authentication phase when the member first joins the group and when the member moves, and the second one is the group key management phase where both the group key and the area group key are updated when members join or leave the multicast session and the updating of the area keys alone are enabled when a member moves.

The CRAW protocol provides a scalable decentralized framework with Independent TEK per subgroup. CRAW reduces the workload of the server as the group members are allowed to generate the required keys. The use of one way hash function and one-time passwords makes CRAW a secured protocol. AWS does not

need to generate individual keys for the members when they move to different areas. The other side of CRAW suffers from 1–affect-n phenomenon within the cluster area which has CKC with a common TEK approach. The movement of the members needs re-authentication during the handoff process which relies on the main server and may be affected due to a single point of failure. The protocol cannot handle the multiple membership change and lacks an efficient multiple authentication mechanism which in turn may affect the main server performance and efficiency.

- **Multicast Key Management Scheme (MKMS).** MKMS, authored by Ming-Chin and Jeng-Farn [43] proposes two multicast key management schemes: LMA based secure group communications and MAG [44] based multicast schemes in decentralized framework Proxy Mobile IPv6 (PMIPv6) [45] networks. This is a modified version of LKH scheme and preserves forward and backward secrecy requirements and also reduces the 1-affects-*n* problem with lower communication costs. PMIPv6, a network based mobility management protocol helps in reducing the handover latency caused by the host based mobility management schemes. PMIPv6 network supports mobile multicast, allowing a Mobile Node (MN) to move from one MAG to another MAG under the same Localized Mobility Domain (LMD) without changing its IP Address. In MAG-based scheme, the MN is allowed to join the multicast group directly and receives multicast messages through the current MAG without passing through LMA and thus reduces the end to end transmission delay compared to the LMA based. In LMA-based scheme the MN joins the multicast group through the LMA and receives the multicast messages through a MAG-LMA tunnel. The movement of MN around the same LMD does not require a re-joining process, as it has joined the multicast session through LMA and hence reduces the joining and handoff delay. However, the LMA-based scheme is suitable for high speed networking environment and the MAG-based scheme is suitable for stable environment. The network entities used are Mobile Access Gateways (MAGs) and Local Mobility Anchor (LMA). MAGs help LMA by signalling about MN movements and also detect the movement of the MN. LMA maintains the binding cache entries for currently registered MNs, Authentication, Authorization, and Accounting (AAA) server, authenticating the MN, Key Distributor Centre (KDC) for generating, distributing and updating the group key and the Service Provider (SP) obtains the group key from the KDC and encrypts the multicast messages and delivers to the MNs either via MAG-based or LMA-based multicast schemes. In LMA based approach, the LMA plays the role of a key node and a member node at the same time. The SP uses the group key from KDC to encrypt the multicast messages and sends it to the LMA. LMA on receiving it decrypts with the group key and encrypts the data again with Domain-GK, and multicasts it to all members via the MAG-LMA tunnel. The members now get access to the multicast messages on decrypting using Domain-GK. The protocol lacks trustworthy relationship which may allow eavesdroppers to get access to multicast messages and cannot handle multiple authentication mechanism. In addition, the tunnel convergence problem may occur when a huge number of members starts to move between the cells.

- **Summary.** A comparison of tree based network dependent protocols based on parameters like key dependence, 1-affects-*n* phenomenon, handling multiple membership changes, scalability, support of security services, fault tolerance and rekey overhead is given in Table 6. Wang scheme is capable of operating in heterogeneous wireless networks by comparing to other protocols that are studied. The centralized nature of the protocol (TMKM, HKM) suffers from a single point of failure and the number of tree levels in the protocol structure also affects the performance in a large manner when more handoffs are to be carried over.

Table 6. Summary of the tree-based Network dependent protocol

Protocol	Key dependence	1-affects-*n*	Multiple membership changes support	Scalability	Security services	Fault tolerant	Rekey overhead
TMKM	Yes	Yes	No	Yes	No	No	Yes
HKM	Yes	Yes	No	Yes	No	No	No
WANG	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CRAW	No	Yes	No	Yes	Yes	No	Yes
MKMS	Yes	Yes (within the cell)	No	Yes	Yes	No	No

4.2. Cluster-based approach

- **Kellil et al. protocol.** K e l l i l, O l i v e r e a u and J a n n e t e a u [46] proposed a decentralized area re-keying algorithms which address member mobility in mobile multicast communication. The mobile members use a specific key called Visitor Encryption Key (VEK). The network entities are Domain Group Controller Key Server (D-GCKS) and Local Group Controller Key Servers (L-GCKS). D-GCKS generates and distribute the Traffic Encryption Keys (TEK) to all L-GCKS of multicast session. L-GCKS forwards the TEK to respective area group members by encrypting with Keys (KEK_i, KEK_j) that are specific to each area (i, j), managed by GCKSs ($GCKS_i$, $GCKS_j$) respectively. Each GCKS maintains two lists, called Extra Key Owner List (EKOL) which holds details about the static members. such as members holding the local area key KEK and Visitor Key Owner List (VKOL) which holds details about moving members, such as members holding VEK, a valid one, but left the area. A member moving from area i to area j, informs about its movement by synchronously sending two signalling messages to $GCKS_i$ area and new $GCKS_j$. The new $GCKS_j$ sends its local VEK_j which acts as a local area key within the area j to the new member via a secure channel. This protocol employs two different methods of a rekeying mechanism for static and mobile members, thus reducing the overhead of area rekeying. Both backward and forward secrecies are assured along the mobility between areas. The protocol uses a common TEK approach and hence it suffers from 1-affects-*n* phenomenon and cannot handle highly dynamic and highly mobile members due to multiple rekey requests.
- **Group Key Management Framework (GKMF).** The main aim of this cluster based network dependent, decentralized group key management protocol is to provide Secure Group Communication in Wireless Mobile Environments.

GKMF, proposed in Kiah and Martin [47, 48] and Kiah and Daghighi [49] uses a common TEK approach. The protocol makes use of lists to manage the dynamic members of the cellular network environment. The network entities are classified as main entities and placement entities. Based on these entities, the participating members or nodes are divided into two levels: a domain level and an area level. The entities at the domain level are called DKM (Domain Key Manager) that generates, distributes, stores, and deletes all the key material that is required at the domain level. The entity at the area level is called AKM (Area Key Manager), the key manager for the area inside a domain. It performs key management in its area and manages its group members. Both the entities maintain a list called MobList to keep track of the mobile members. The MobList contains: moving member ID, multicast group G, joined by the member, the area that a member is moving from, ID of the target area that a member is moving to. The list is updated for every handoff process which can be used to track the details of the mobile members and for a rekeying process in the newly visited areas. GKMF uses shared symmetric keys to provide secure association at different levels and thus assures a trustworthy relationship between the communicating entities. The other side of the protocol suffers from storage overhead, since large numbers of used keys are needed to be stored. The protocol does not assure forward secrecy for the protocol does not address the re-keying mechanism in the area the member is leaving and also suffers from 1-affect-n phenomenon due to a common TEK. The protocol assures backward secrecy, but causes a joining delay as area rekeying and TEK rekeying are carried out independently.

- **Gharout et al. protocol.** Gharout et al. [50, 51] proposed a new key management protocol aimed to provide a secure group communication in mobile network environment with null rekeying cost. The protocol uses independent TEK per subgroup and thus overcomes 1-affect-n phenomenon. The network entities are the Domain Key Distributors (DKD) which manages all the Area Key Distributors (AKDs) under its control and AKDs performs key management for its area and also authenticates the mobile members. The participating members are grouped into clusters with each cluster controlled by one DKD at the domain level and at least one AKD at the area level. AKDs under the same DKD use a common TEK approach and no rekeying is required when a member moves within the same cluster and thus it optimizes the rekeying process. The participating members are at the area level. Each AKD has two lists, namely a list containing currently present members in its area and a list, containing details of previous members that have left its area. This approach assures forward and backward secrecy services. The mobile members participating in multiple sessions need to store multiple encryption keys, thus resulting in storage overhead. The details of rekeying when the members move between two different clusters are not addressed in this approach.

- **Key Management to secure Group communications in Mobile environments (KMGM).** KMGM proposed by Chung Kei, Gouda and Lam [52] improves the performance of Adaptive clustering for scalable key management in dynamic group communications protocol by adding mobility support for the multicast members in mobile network environment. The protocol addresses both

intra cluster and inter cluster mobility. KMGM follows a hybrid approach by using both the common TEK and Independent TEK per subgroup approaches. The decentralized nature of this protocol allows the members to be arranged in a hierarchy of administrative areas, which are managed by Area Key Distributors (AKDs). The AKDs within the cluster are considered to be passive and do no data transformation. The passive AKDs just receive and forward the messages to their respective area members. Both the active AKDs and passive AKDs maintain two lists: a list of the currently present members of its area and a list of the old members in this area. The operational performance of KMGM is somewhat similar to Gharout et al. protocol, but KMGM addresses the rekeying details when the members move between two different clusters (inter clusters).

- **Summary.** Table 7 summarizes the cluster based network dependent protocols based on the key characteristics of the protocol. K e l l i l, O l i v e r e a u and J a n n e t e a u [46] and GKMF suffer from a single point of failure.

Table 7. Summary of cluster based network dependent protocols

Protocol	Key Depen-dence	1-affects-*n*	Multiple membership changes support	Calability	Security services	Fault tolerant	Rekey overhead
Kellil et al.	Yes	Yes	No	Yes	No	No	Yes
GKMF	Yes	Yes	No	Yes	No	No	Yes
Gharout et al.	Yes (intra cluster)	Yes (intracluster)	No	Yes	Yes	Yes	No
KMGM	Yes (intra cluster)	Yes (intra cluster)	No	Yes	Yes	Yes	No

5. Conclusion

The paper discusses various approaches of group key management both in network independent environment and network independent environment. The survey clearly shows that each protocol, following various approaches like centralized, decentralized and distributed framework, has its unique features. The centralized approach is easy to implement. The decentralized framework provides a scalable structure by dividing the participating group members into sub groups. The distributed framework allows every participating member to take part in the key management activities.

The success of multicast communication relies on the security of TEK used. Thus an efficient group key management is required to generate, distribute and update the group key in a secure manner over unsecured channel. The survey discusses the use of a common TEK approach and independent TEK approach per sub group. To propose efficient key management protocol characteristics like delay, 1-affect-n phenomenon, storage overhead, rekey overhead, computational cost has to be highly considered. In cellular network environment [55], the resource constraints, bandwidth constraints, highly dynamic environment, highly changing membership must be considered for efficient key management and successful multicast communication.

References

1. Judge, P., M. Ammar. Security Issues and Solutions in Multicast Content Distribution: A Survey. – Network, IEEE, Vol. **17**, 2003, pp. 30-36.
2. Challal, Y., H. Seba. Group Key Management Protocols: A Novel Taxonomy. – Enformatika, International Journal of Information Technology, Vol. **2**, 2005.
3. Bibo, J., H. Xiulin. A Survey of Group Key Management. – In: Computer Science and Software Engineering, 2008 International Conference on, 2008, pp. 994-1002.
4. Rafaeli, S., D. Hutchison. A Survey of Key Management for Secure Group Communication. – ACM Computing Surveys, Vol. **35**, September 2003, pp. 309-329.
5. Chiou, G. H., W. T. Chen. Secure Broadcast Using Secure Lock. – IEEE Transactions on Software Engineering, Vol. **15**, August 1989, No 8, pp. 929-934.
6. Harney, H., C. Muckenhirn. Group Key Management Protocol (GKMP) Architecture, July 1997. – In: RFC 2093.
7. Harney, H., C. Muckenhirn. Group Key Management Protocol (GKMP) Specification, July 1997. – In: RFC 2094.
8. Wallner, D., E. Harder, R. Agee. Key Management for Multicast: Issues and Architecture. National Security Agency, June 1999. – In: RFC 2627.
9. Wong, C. K., M. Gouda, S. S. Lam. Secure Group Communications Using Key Graphs. – In: Proc. of ACM SIGCOMM, 1998.
10. Wong, C. K., M. Gouda, S. S. Lam. Secure Group Communications Using Key Graphs. – IEEE/ACM Transactions on Networking, Vol. **8**, February 2000, No 1, pp. 16-30.
11. Balenson, D., D. McGrew, A. Sherman. Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization. Draft-balenson-groupkeymgmtoft-00.txt, February 1999. Internet-Draft.
12. Canetti, R., J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas. Multicast Security: A Taxonomy and Efficient Constructions. – In: Proc. of IEEE INFOCOM, March 1999, pp. 708-716.
13. McGrew, D. A., A. T. Sherma. Key Establishement in Large Dynamic Groups Using One-Way Function Trees. Technical Report TR-0755, World Academy of Science Engeneering and Technology, May 1998.
14. Waldvogel, M., G. Caronni, D. Sun, N. Weiler, B. Plattner. The VersaKey Framework: Versatile Group Key Management. – IEEE Journal on Selected Areas in Communications (Special Issues on Middleware), Vol. **17**, August 1999, No 8, pp. 1614-1631.
15. Ballardie, A. Scalable Multicast Key Distribution. May 1996. – In: RFC 1949.
16. Ballardie, A. Core Based Trees (CBT Version 2) Multicast Routing Protocol Specification, September 1997. – In: RFC 2189.
17. Ballardie, T., I. P. Francis, J. Crowcroft. Core Based Trees: An Architecture for Scalable Inter-Domain Multicast Routing. – In: Proc. of ACM SIGCOMM, 1993, pp. 85-95.
18. DeCleene, B., L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, C. Zhang. Secure Group Communications for Wireless Networks. MILCOM, June 2001.
19. Hardjono, T., B. Cain, I. Monga. Intra-Domain Group Key Management for Multicast Security. IETF Internet Draft, September 2000.
20. Rafaeli, S., D. Hutchison. Hydra: A Decentralized Group Key Management. – In: 11th IEEE International WETICE: Enterprise Security Workshop, June 2002.
21. Setia, S., S. Koussih, S. Jajodia, E. Harder. Kronos: Ascalable Group Re-Keying Approach for Secure Multicast. – In: IEEE Symposium on Security and Privacy, May 2000.
22. Briscoe, B. MARKS: Multicast Key Management Using Arbitrarily Revealed Key Sequences. – In: 1st International Workshop on Networked Group Communication, November 1999.
23. Dondeti, L. R., S. Mukherjee, A. Samal. Scalable Secure One-to-Many Group Communication Using Dual Encryption. – Computer Communications, Vol. **23**, November 2000, No 17, pp. 1681-1701.

24. D o n d e t i, L. R., S. M u k h e r j e e, A. S a m a l. Comparison of Hierarchical Key Distribution Schemes. – In: IEEE Globcom Global Internet Symposium, 1999.
25. I n g e m a r s o n, D. T a n g, C. W o n g. A Conference Key Distribution System. – IEEE Transactions on Information Theory, Vol. **28**, September 1982, No 5, pp. 714-720.
26. S t e i n e r, M., G. T s u d i k, M. W a i d n e r. Diffie-Hellman Key Distribution Extended to Group Communication. – In: 3rd ACM Conference on Computer and Communications Security, March 1996, pp. 31-37.
27. D i f f i e, W., M. E. H e l l m a n. New Directions in Cryptography. – IEEE Transactions on Information Theory, Vol. **IT-22**, November 1976, pp. 644-654.
28. B e c k e r, C., U. W i l l e. Communication Complexity of Group Key Distribution. – In: 5th ACM Conference on Computerand Communications Security, Nowember 1998.
29. K i m, Y., A. P e r r i g, G. T s u d i k. Communication-Efficient Group Key Agreement. – In: Proc. of IFIP SEC, June 2001.
30. R o d e h, O., K. B i r m a n, D. D o l e v. Optimized Group Rekey for Group Communication Systems. – Network and Distributed System Security, February 2000.
31. D o n d e t i, L., S. M u k h e r j e e, A. S a m a l. A Distributed Group Key Management Scheme for Secure Many-to-Many Communication. Technical Report PINTL-TR-207-99, 1999.
32. F i a t, A., M. N a o r. Broadcast Encryption. – In: CRYPTO'93, 1993, LNCS(773), pp. 480-491.
33. B o y d, C. On Key Agreement and Conference Key Agreement. – In: Information Security and Privacy: Australasian Conference, 1997, LNCS(1270), pp. 294-302.
34. B u r m e s t e r, M., Y. D e s m e d t. A Secure and Efficient Conference Key Distribution System. – In: EUROCRYP'94, 1994, LNCS(950), pp. 275-286.
35. Y a n, S., W. T r a p p e, K. J. R. L i u. An Efficient Key Management Scheme for Secure Wireless Multicast. – In: ICC'02. Communications, 2002, IEEE International Conference on, Vol. **2**, 2002, pp. 1236-1240.
36. Y a n, S., W. T r a p p e, K. J. R. L i u. Topology-Aware Key Management Schemes for Wireless Multicast. – In: Global Telecommunications Conference, 2003. GLOBECOM'03, Vol. **3**, IEEE, 2003, pp. 1471-1475.
37. L i n, L., L. X u e m i n g, C. Y o n g. HKM: A Hybrid Key Management Scheme for Secure Mobile Multicast. – In: Networking, Architecture, and Storage, 2007. NAS'2007. International Conference on, pp. 109-114.
38. Y i l i n g, W., L. P h u D u n g, B. S r i n i v a s a n. Hybrid Group Key Management Scheme for Secure Wireless Multicast. – In: Computer and Information Science, 2007. ICIS'2007. 6th IEEE/ACIS International Conference on, 2007, pp. 346-351.
39. Y i l i n g, W., L. P h u D u n g, B. S r i n i v a s a n. Efficient Key Management for Secure Wireless Multicast. – In: 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT'08), 2008., pp. 1131-1136.
40. E i d k h a n i, E., M. H a j y v a h a b z a d e h, S. A. M o r t a z a v, A. N. P o u r. CRAW: Combination of Re-Keying and Authentication in Wireless Networks for Secure Multicast Increasing Efficiency of Member Join/Leave and Movement. – International Journal of Computer Networks & Communications (IJCNC), Vol. **4**, 2012, pp. 107-128.
41. S a n d i r i g a m a, M., S. A k i h i r o, M. N o d a. Simple and Secure Password Authentication Protocol. – IEICE Trans. Com., Vol. **E83-B**, 2000, pp. 1363-1365.
42. H a j y v a h a b z a d e h, M., E. E i d k h a n i, S. A. M o r t a z a v i, A. N. P o u r. A New Group Key Management Protocol Using Code for Key Calculation: CKC. – In: International Conference on Information Science and Applications (ICISA'10), 2010, pp. 1-6.
43. M i n g-C h i n, C., L. J e n g-F a r n. MKMS: Multicast Key Management Scheme for Proxy Mobile IPv6 Networks. – In: International Conference on Consumer Electronics, Communications and Networks (CECNet'11), 2011, pp. 1402-1405.
44. J i a n f e n g, G., Z. H u a c h u n, Z. H o n g k e, H. L u o. Multicast Extension Support for Proxy MIPv6. – In: Consumer Communications and Networking Conference (CCNC'10), 7th IEEE, 2010, pp. 1-5.
45. G u n d a v e l l i, S., K. L e u n g, V. D e v a r a p a l l i, K.C h o w d h u r y, B. P a t i l. Proxy Mobile IPv6. RFC 5213, August 2008.
46. K e l l i l, M., J. C. A. O l i v e r e a u, P. J a n n e t e a u. Rekeying in Secure Mobile Multicast Communications. United States Patent Application Publications, US 2007/0143600 A1 2007.

47. K i a h, L. M., K. M. M a r t i n. Host Mobility Protocol for Secure Group Communication in Wireless Mobile Environments. – In: Future Generation Communication and Networking (FGCN'07), 2007, pp. 100-107.
48. K i a h, M. L. M., K. M. M a r t i n. Host Mobility Protocol for Secure Group Communication in Wireless Mobile Environments. – International Journal of Security and its Applications, Vol. **2**, January 2008, pp. 39-52.
49. K i a h, M. L. M., B. D a g h i g h i. An Implementation of Secure Group Communication in a Wireless Environment. – International Journal of Computer and Electrical Engineering, Vol. **4**, December 2012.
50. G h a r o u t, S., A. B o u a b d a l l a h, M. K e l l i l, Y. C h a l l a l. Key Management with Host Mobility in Dynamic Groups. – In: Proc. of 3rd International Conference on Security of Information and Networks, Taganrog, Rostov-on-Don, Russian Federation, 2010.
51. G h a r o u t, S., A. B o u a b d a l l a h, Y. C h a l l a l, M. A c h e m l a l. Adaptive Group Key Management Protocol for Wireless Communications. – j-jucs, Vol. **18**, May 2012, pp. 874-898.
52. C h u n g K e i, W., M. G o u d a, S. S. L a m. Secure Group Communications Using Key Graphs. –IEEE/ACM Transactions on Networking, Vol. **8**, 2000, pp. 16-30.
53. M a p o k a, T r u s t T s h e p o. Group Key Management Protocols for Secure Mobile Multicast Communication: A Comprehensive Survey. – International Journal of Computer Applications, Vol. **84**, 2013, pp. 28-38.
54. Z h a o, S., R. K e n t, A. A g g a r w a l. A Key Management and Secure Routing Integrated Framework for Mobile Ad Hoc Networks. – Ad Hoc Networks, Vol. **11**, 2013, pp. 1046-1061.
55. A b o u h o g a i l, R e h a m A b d e l l a t i f. Security Assessment for Key Management in Mobile Ad Hoc Networks. – International Journal of Security and Its Applications, Vol. **8**, 2014, No 1, pp. 169-182.