Cluster Based Group Key Management in Mobile Ad hoc Networks

Renuka A.[†] and K.C.Shet[††],

Dept. of Computer Science & Engineering., M.I.T., Manipal, Manipal University-576104 (India)
[††] Dept. of Computer Engineering, N.I.T.K., Surathkal-575025 (India)

Summary
A Mobile Ad-hoc Network (MANET) is a collection of autonomous nodes or terminals which communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. We propose a decentralized cluster based group key management scheme that uses a symmetric group key for communication within the cluster. This group key is generated by the cluster head and communicated to other members through a secure channel that uses public key cryptography. We show through simulations that clustering reduces the packet losses compared to the centralized scheme. We also discuss the method for authentication of public keys. We analyze the performance of our scheme and compare it with the existing schemes.

Key words:
authentication, group key, hash tree, public key cryptography

1. Introduction

An ad hoc network is a collection of autonomous nodes that communicate with each other, most frequently using a multi-hop wireless network. Nodes do not necessarily know each other and come together to form an ad hoc group for some specific purpose. Key distribution systems usually require a trusted third party that acts as a mediator between nodes of the network. Ad hoc networks typically do not have an online trusted authority but there may be an off line one that is used during system initialization. A node in an ad hoc network has direct connection with a set of nodes, called neighboring nodes, which are in its communication range. The number of nodes in the network is not necessarily fixed. New nodes may join the network while existing ones may be compromised or become un-functional.

Group key establishment means that multiple parties want to create a common secret to be used to exchange information securely. Without relying on a central trusted entity, two people who do not previously share a common secret can create one based on the DH protocol. The 2-party Diffie Hellman (DH) protocol can be extended to a generalized version of n-party DH. Research efforts have been put into the design of a group key management scheme for the sake of scalability, reliability, and security. Furthermore, group key management also needs to address the security issue related to membership changes. The modification of membership requires refreshment of the group key. This can be done either by periodic rekeying or updating right after member change. The change of group key ensures backward and forward security. With frequently changing group memberships, recent researches began to pay more attention on the efficiency of group key update. Group key management protocols can be roughly classified into three categories; centralized, decentralized, and distributed [1]. In centralized group key protocols, a single entity is employed to control the whole group and is responsible for group rekeying. In the decentralized approach, multiple entities are responsible for managing the group as opposed to a single entity. In the distributed method, group members themselves contribute to the formation of a group key and are equally responsible for the rekeying and distribution of group keys. Recently, collaborative and group-oriented applications in MANETs have been an active research area. Obviously, group key management is a central building block in securing group communications in MANETs. However, group key management for large and dynamic groups in MANETs is a difficult problem because of the requirement of scalability and security under the restrictions of nodes' available resources and unpredictable mobility.

We propose a decentralized group key management approach wherein there is no central authority and uses a combination of public and private key cryptography. We treat the total network topology as a group and every cluster as a subgroup. We assume in every cluster, every node can receive a message broadcasted from the other nodes. The entire set of nodes in the network is divided into a number of clusters. Each cluster is headed by a cluster head. The nodes within the cluster communicate using symmetric key cryptography with the group key shared by all the nodes. Asymmetric key cryptography is used to encrypt the group keys generated, whenever membership changes occur. The authentication is done using hash trees or authentication trees. Even cluster heads authenticate each other using separate hash trees. By this approach, we can minimize the computation cost of

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

authentication and communication costs of rekeying messages whenever membership changes occur. This will also eliminate the requirement of a centralized authority for generating and distributing the keys. We also analyze the performance of the scheme whenever a member joins or leaves.

The rest of the paper is organized as follows. Section 2 focuses on the related work in this field. The proposed scheme is presented in Section 3. Experimental Setup is discussed in Section 4. Performance analysis of the scheme and conclusion are given in Section 5 and Section 6 respectively

2. Related Work

Key management is a basic part of any secure communication. Most cryptosystems rely on some underlying secure, robust, and efficient key management system. Secure network communications normally involve a key distribution procedure between communication parties, in which the key may be transmitted through insecure channels. A framework of trust relationships needs to be built for authentication of key ownership in the key distribution procedure. While some frameworks are based on a centralized trusted third party (TTP), others could be fully distributed. For example, a certification authority (CA) is the TTP in asymmetric cryptosystems, a key distribution center (KDC) is the TTP in the symmetric system, and in pretty good privacy (PGP) no TTP is assumed. The centralized approach is regarded as inappropriate for MANETs because of the dynamic environment and the transient relationships among mobile nodes

Key distribution and key agreement over an insecure channel are at high risk and suffer from potential attacks. In the traditional digital envelop approach, a session key is generated at one side and is encrypted by the public-key algorithm. Then it is delivered and recovered at the other end. In the Diffie-Hellman (DH) scheme [2], the communication parties at both sides exchange some public information and generate a session key on both ends. Several enhanced DH schemes have been invented to counter man-in-the-middle attacks. In systems lacking a TTP, the public-key certificate is vouched for by peer nodes in a distributed manner, such as pretty good privacy (PGP) [2]. .

Group key establishment means that multiple parties want to create a common secret to be used to exchange information securely. Without relying on a central trusted entity, two people who do not previously share a common secret can create one based on the DH protocol. The 2-party Diffie Hellman (DH) protocol can be extended to a generalized version of n-party DH. Research efforts have been put into the design of a group key management scheme for the sake of scalability, reliability, and security. Furthermore, group key management also needs to address the security issue related to membership changes. The modification of membership requires refreshment of the group key. This can be done either by periodic rekeying or updating right after member change.

Logical Key Hierarchy (LKH) is a centralized group key management scheme proposed by Wallner, Harder and Agee [9]. It is based on the tree structure with each user (group participant) corresponding to a leaf and the group initiator as the root node. The tree structure will significantly reduce the number of broadcast messages and storage space for both the group controller and group members. Each leaf node shares a pair wise key with the root node as well as a set of intermediate keys from it to the root. So, for a balanced binary tree, each group member stores at most d+1 keys, where, d is the height of the tree, and n is the total number of group members. One Way Function (OFT) is another centralized group key management scheme proposed by Sherman and McGrew [3]. It is based on the tree structure that is similar to the above LKH scheme. However, all keys in the OFT scheme are functionally related according to a one-way hash function. A group user still needs to store d+1 keys, where d is the height of the tree, and n is the total number of group members. The scheme has the same complexity as the LKH scheme for a balanced tree structure, but in the re-keying process, the size of keying materials reduces from 2logn to log n . Tree Based Group Diffie Hellman (TGDH) is a group key management scheme proposed in [4]. The basic idea is to combine the efficiency of the tree structure with the contributory feature of DH. The basic operation of this scheme is as follows. Each group member contributes its (equal) share to the group key, which is computed as a function of all the shares of current group members. As the group grows, new members' shares are factored into the group key but old members' shares remain unchanged. As the group shrinks, departing members' shares are removed from the new key and at least one remaining member changes its share. All protocol messages are signed by the sender using RSA. In Simple and Efficient Group Key (SEGK) management scheme for MANETs proposed in [5] group members compute the group key in a distributed manner. The basic idea of the scheme is that a multicast tree is formed in MANETs for efficiency. Group members take turns to act as a group coordinator to compute and distribute the intermediate keying materials to all members through the active tree links. Also MS. Bouassida, I. Chrisment, and O. Festor [6] developed a new approach, called BALADE, based on a sequential multi-sources model, and takes into account both localization and mobility of nodes, while optimizing energy and bandwidth consumptions.

Group communication is usually done using symmetric key cryptography and distribution of group key is done through (Public Key Cryptography). In this paper, we propose the method in which symmetric key is used for communication between the nodes of a cluster and asymmetric key cryptography for distributing the group key to the members of the cluster. The authentication of public keys is done using authentication trees [7]. This eliminates the necessity of having a certification authority or a centralized authority for authenticating the sender of the group key.

3. Proposed Scheme

3.1 System Model

The entire set of members in the network is divided into a number of subsets called clusters. Each cluster is headed by a cluster head. The layout of the network is as shown in Figure1.The cluster head is similar to other members with same computational capability as other members in the network. The communication encrypted with the public keys forms a secure channel for communication between the nodes in the cluster. Hash trees or authentication trees are used for authentication of public keys. The cluster head generates a group key and distributes it to its members through the secure channel. Whenever membership changes occur, the cluster head regenerates the group key and distributes through this secure channel. One among the cluster heads is elected as the leader for managing the key generation and distribution of the cluster heads. Though this scheme looks like the centralized scheme, this is actually a decentralized one, as there is no central point of failure. The entire protocol can restart without affecting the security of the network.

3.2 Assumptions

1. Each node in the mobile ad hoc network is assigned a unique integer valued id for identifying the node. The status code for the cluster heads is 1 and for the members is 0.
2. We assume that there is an offline authority that deploys the nodes into the network and issues a valid certificate binding the id with the public keys for authenticating the member, when it initially enters the network.

3. There are two types of keys that are used within a cluster.
(i) Group key GK, used for encryption of messages exchanged in group communication within the cluster
(ii)Public key pk, of individual members used as Key encrypting key (KEK), for encrypting the group key GK generated whenever membership changes occur

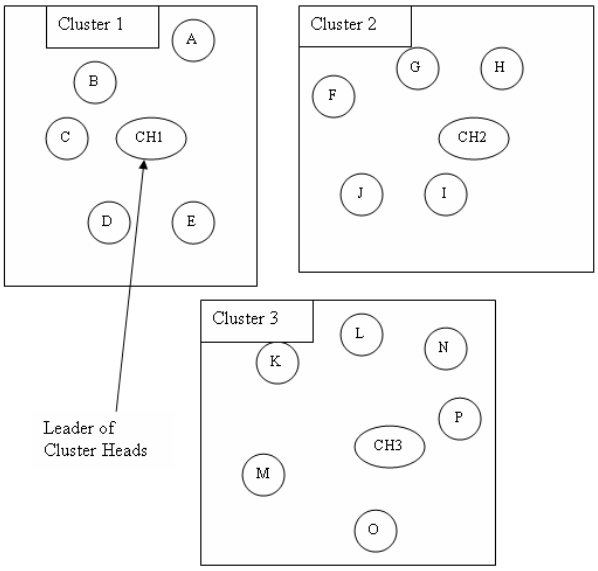

Fig. 1 Network Layout

The protocol consists of different phases.
1. Initialization
2. Group Key Generation and Distribution

3.3 Initialization

Step 1: After deployment, the nodes broadcast their id value to their neighbors along with the HELLO message.
Step 2: The node with the smallest id among all the neighbors is selected as the cluster head. The nodes update the status values accordingly.
Step 3: The cluster head broadcasts the message “I am cluster head” so as to know its members.
Step 4: The members reply with the message “I am member” and in this way clusters are formed in the network.
Step 5: If a node receives more than one “I am cluster head” messages, based on the received signal strength it chooses the node with the larger signal strength as its cluster head.
In this manner disjoint clusters are formed in the network

3.4 Group Key Generation and Distribution

This phase consists of two parts

Group key generation and distribution within the cluster
Group key generation and distribution between cluster heads

3.4.1Group key generation and distribution within the cluster

Step 1: The cluster head broadcasts its public key to its members.
Step 2: The members broadcast their public keys along with their ids to all the members in the cluster head.
Step 3: The cluster head constructs the hash tree with the members and itself as the leaves of the tree. This is explained below
The hash tree is balanced if the number of members is a power of two as shown in Figure 3. Otherwise an unbalanced tree is constructed as shown in Figure 4.
The values of the leaves are obtained as
$i=hash(\ id_i\ ||\ pk_i\ )$ for i=1,…M
Where idi is used to represent the node i's identity, and pk_i to represent node's public key. The function hash is a one-way hash function such as MD5 or SHA1. The values of the intermediate nodes are obtained by hashing the concatenated value of its two children. For e.g. the value D is obtained as
D=hash(1 || 2).
This is repeated until the root is reached. The root value is called the root hash and is to be stored by all the members for authentication purposes.

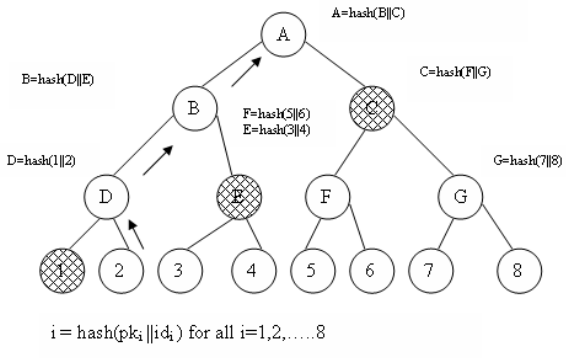

Fig. 2. Balanced Hash Tree

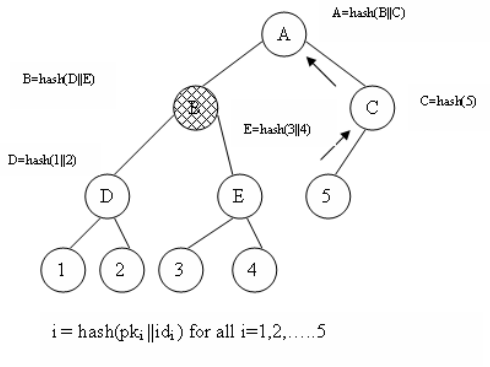

Fig.3. Unbalanced Hash Tree

To authenticate a particular leaf node, the path from the leaf to the root is traced to arrive at the root hash value. For e.g. in Figure 2, node 2 traces the path D, B to the root. To enable the destination node properly compute the root hash value, for each intermediate node in the tree the node has to send the sibling value called the proofs to the destination. Therefore node 2 it has to send the 1, E and C. The destination node computes the root hash value from the hash values received and compare with the stored root hash value. If both match, the leaf node is an authentic node. In case of an unbalanced hash tree, the same procedure is followed, but the number of proofs to be sent reduces for certain nodes. For e.g. in Figure 3, leaf node 5 just sends value B to authenticate itself.
Step 4: The cluster head sends the root hash value to each of its members. It also sends the intermediate hash values to each of the nodes so that the nodes can authenticate their public keys.
Step 5: The cluster head generates a symmetric key using AES (Advanced encryption Standard), called the group key GK and sends it to each of the members encrypted with their corresponding public key along with the hash values on its path to the root for authentication.

3.4.2 Group key generation and distribution between cluster heads

Step 1: The cluster heads broadcast the message "Are there any other cluster heads" along with its id and public key and collects information regarding the cluster heads.
Step 2: All the cluster heads construct the hash tree with itself and the other cluster heads as the leaves and compute the root hash value. This root hash value is used for authentication among the cluster heads.
Step 3: The cluster heads then choose the cluster head with the smallest id as their leader.

Step 4: The leader of the cluster heads generates a new group key GK_{CH} which is then used for communication among the cluster heads.
The notations used in this paper are listed in Table1

Table 1:.Notations

Symbol	Meaning
GK_i	Group Key of cluster i
$E_{GKi}[m]$	Encyption of message m by GK_i
$D_{GKi}[E_{GKi}[m]]$	Decryption of message m encrypted by GK_i using GK_i
GK_{CH}	Group key of cluster heads
M	Number of members in a cluster
N	Total number of members in the network
P	No. of clusters in the network
$h=Log_2 M$	Height of the hash tree for a cluster
hash	Hash function such as SHA-1 or SHA-256

3.5 Communication Protocol

3.5.1 Intra cluster Communication

The members within the cluster communicate using the group key. Suppose member C wants to send a message to member D in the same cluster it encrypts the message with the group key GK_1 as shown in Figure 4.
C→ D : $E_{GK1}[m]$
D→ $D_{GK1}[E_{GK1}[m]]$

3.5.1 Inter cluster Communication

The members in one cluster communicate with the members in other cluster through their respective cluster heads. Suppose member A in cluster 1 wants to communicate with member I in cluster2 it first sends the message to its cluster head CH_1. The cluster head decrypts the message, and then encrypts the message using the cluster head group key GK_{CH} and sends it to the parent cluster head CH_2 of member I. The cluster head of CH_2 member I again decrypts the message and encrypts with its own group key and sends it to I.
A: $E_{GK1}[m] \rightarrow CH_1$
$CH_1 \rightarrow CH_2$: $EGK_{CH}[D_{GK1}[E_{GK1}[m]]] \rightarrow CH_1$
$CH_2 \rightarrow I$: $E_{GK2}[D_{GKCH}[E_{GKCH}[D_{GK1}[E_{GK1}[m]]]]]$
I: $D_{GK2}[E_{GK2}[D_{GKCH}[E_{GKCH}[D_{GK1}[E_{GK1}[m]]]]]]$

This requires a total of three encryptions and three decryption operations.

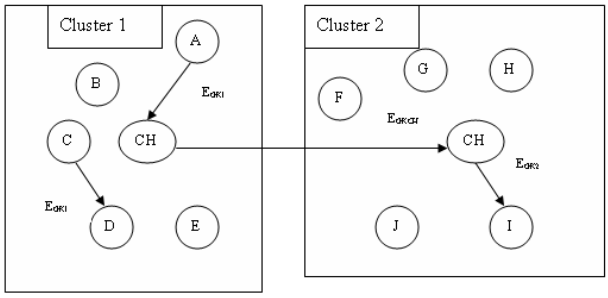

Fig 4 Illustration of Communication

3.6 Network Dynamics

The number of nodes in the network is not necessarily fixed. New nodes may join the network or existing nodes may leave the network. Each node in a MANET may not be fixed in one position and may sometimes move frequently. In this section we address these issues

3.6.1 Member Joins

Step 1: When a new member joins the network, it sends a join request message to the adjacent node.
Step 2: After receiving the join request message, the adjacent node sends this message to the cluster head using the group key.
Step 3: The cluster head sends a reply to the new node after which the new node joins the cluster.
Step 4: The new member then sends its public key to the cluster head.
Step 5: The cluster head reconstructs the hash tree along with the new member as the leaf of the hash tree and sends the root hash value to the new member and broadcasts it to the other nodes encrypted with the old group key. It also computes the new group key and unicasts it to the new member and broadcasts the same to the old members again encrypting with the old group key

3.6.2 Member Leaves

When a node leaves there are three cases
Case (i): Cluster member leaves
Case (ii): Cluster head leaves
Case (iii): Leader of the cluster heads leaves

3.6.2.1 Cluster Member Leaves

Step 1: Before leaving, the member informs the cluster head that it is leaving.

Step 2: The cluster head gives permission to the leaving member to leave.
Step 3: The cluster head computes a new group key and distributes it to the members using unicast message encrypted with the members' public key. This is required to ensure that the leaving member is not able to access the new group key.
Step 4: The cluster head reconstructs the hash tree without the leaving member and obtains the root hash value. These hash values are sent to the individual members encrypted with their respective public keys.

3.6.2.2 Cluster Head Leaves

Step 1: Before leaving the cluster, the cluster head sends a leave message to its own members and the other cluster heads.
Step 2: After receiving the leave message, the ordinary nodes and the cluster heads send a reply message to the leaving cluster head. If the cluster head does not receive the reply message in the period, it will send the leave message again to irresponsive members.
Step 3: The cluster head delegates the role of the cluster head to the member that is having the smallest id and transfers all the information to this node.
Step 4: The new cluster head reconstructs the new hash tree for authentication among its members and also generates a new group key and sends it individually to the members encrypted with their public keys. This is very much similar to the process of a cluster member leaving.
Step 5: The cluster heads remove the leaving cluster heads entry and reconstruct the hash tree without the leaving cluster head and compute the new root value for authentication purpose.
Step 6: The cluster head with the lowest id regenerates the new group key and sends it the cluster heads encrypted with their respective public keys.
Step 7: When the cluster head is compromised and not leaving voluntarily, the members in the cluster communicate with each other and elect the node having the smallest id as the cluster head and the whole process of initialization is repeated.

3.6.2.3 Leader of Cluster Heads Leaves
Step 1: The leader informs the other cluster heads that it is leaving
Step 2: The other cluster heads delete the entry of the leaving cluster head and reconstruct the hash tree to compute the root hash value
Step 3: The cluster heads elect the new leader as one with the smallest id.
Step 4: The new leader computes the new group key value and sends it to the other cluster heads

4 Performance Analysis

4.1 Security analysis

Members in a mobile ad hoc network are usually considered to be part of the security issue since there are no fixed nodes to perform the service of authentication. We assume that all of the incoming members of the MANET carry a valid certificate issued by an offline authority binding the node id with its public key.

4.1.1 Forward Secrecy

When a new member joins we ensure that it is not able to receive the previous information that was exchanged prior to it joining the network. In our scheme, the cluster head will regenerate subgroup key and broadcasts it to the old members encrypted with the old group key. It also unicasts the new group key to the new member. Therefore, the new member can encrypt and decrypt later information, but cannot decrypt the previously exchanged information

4.1.2 Backward Secrecy

When a node leaves, we have to make sure that node should not receive any information from the network. We discuss the node leaving by the following three cases.
Case (i): When cluster member leaves:
A new group key GK is regenerated by the cluster head and sent to the members in the cluster. The hash values required for authentication are changed.
Case (ii): When cluster head leaves:
The cluster members not only select a new cluster head, but also regenerate subgroup key for the cluster. The new subgroup key is sent to all members in the cluster headed by the new cluster head. Then the cluster head with the smallest id regenerates the group key for the cluster heads and sends it to all the cluster heads encrypted with their respective public keys. The hash tree for the cluster heads is reconstructed and the hash values required for authentication are changed.
Case (iii): When leader of the cluster head leaves:
In this case, the new leader is chosen by selecting the cluster head having the lowest id and new group key for the cluster heads is regenerated and sent to the other cluster heads in a secure manner by encrypting with their public keys. The hash tree for the cluster heads is reconstructed and the hash values required for authentication are changed
This ensures that the backward secrecy is maintained and the group keys are communicated in a secure manner

4.1.3 Node Capture

When a member is captured by an adversary, there is no threat to the security as only the group key is compromised

which can be regenerated by the cluster head and distributed to the other members in a secure manner. This is similar to the case of member leaving. When the cluster head or the leader of the cluster heads is compromised, security mechanism in only a part of the network is affected and can be rectified within a short span of time, whereas in centralized schemes, if the central key distribution centre itself is compromised, the security of the entire network is at stake. This indicates that our scheme is robust against attacks.

4.2 Time Complexity and Storage Cost Analysis

The comparison of the time complexity and storage cost of our approach with the efficient and commonly used centralized schemes is given in Table2.

4.2.1 Member Joins

Suppose the cluster consists of M nodes, the number of changes in the hash values is Log_2M or h where h is the height of the hash tree (worst case). It is just sufficient to change only those hashes along the path from the point of insertion to the root which involves a computational complexity of less than Log_2M. The communication complexity is one broadcast message and one unicast message. The group keys of other clusters need not be changed.

4.2.2 Member Leaves

When a node leaves, there are three cases
(i) The cluster member leaves
(ii)The cluster head leaves
(iii)The leader of the cluster heads leaves

(i)When the cluster member leaves, the hash tree of the cluster to which the member was attached needs to be changed, which involves the computation cost of Log_2M and a communication cost of M-1. Since the old group key is known to the leaving member, these hashes have to be sent individually to the members encrypted with their public keys. The new group key is computed and sent individually to M-1 members.

Table 2: Time Complexity and Storage Cost

	Proposed Scheme	**LKH scheme**	**OFT**
No. of keys stored in the leader of cluster heads	O(M)+O(P)	-----	----
No. of keys stored in the cluster heads	O(M)	----	----
No .of keys stored in the members	O(1)	2N-1	
Communication cost of rekeying in member join	O(1)	O(2logN) =O(2log(MP))	O(logN)
Communication cost of rekeying when cluster member leaves	O(M)	O(2logN) =O(log(MP))	O(logN)
Communication cost of rekeying when cluster head leaves	O(M)+O(P)	---	----
Communication cost of rekeying when leader of cluster heads leaves	O(M)+O(P)	---	----
Robust against attacks	Yes	No	No

(ii)When the cluster head voluntarily leaves, it delegates the responsibility of the cluster head to another node, by transferring the information about its members and the other cluster heads to the newly selected cluster head. The new cluster head then reconstructs the hash tree and computes the group key and sends it to other N/M cluster heads. The hash tree of the cluster heads also has to be changed. This involves a computational complexity of $Log_2(N/M)$ and no communication cost as this is computed by all the cluster heads.

5 Experimental Setup

The simulations are performed using Network Simulator (NS-2) [8], particularly popular in the ad hoc networking community. The MAC layer protocol IEEE 802.11is used in all simulations. The Ad Hoc On-demand Distance Vector (AODV) routing protocol is chosen for the simulations. Every simulation run is 1000 seconds long. The simulation is carried out using different number of nodes. The simulation parameters are shown in Table 3.

Table 3: Simulation Parameters

Simulation time	1000 sec
Topology size	1000m X 1000m
No. of nodes	200, 80 ,32, 16
No. of clusters	8,4,2
Node mobility	0 to 20m/sec
Routing Protocol	AODV
Frequency	11 MHz
Traffic type	CBR
MAC	IEEE 802.11
Mobility model	Random Waypoint
Max. no. of packets	10000
Pause time	10sec

The experiments are conducted with network sizes of 200 nodes, 80 nodes and 32 nodes. With a network size of 200 nodes, we formed 2, 4, and 8 clusters with 100, 50 and 25 nodes respectively and calculated the packet loss and average end to end delay. This is repeated for network sizes of 80 and 32 and 16 nodes. We compared the packet loss and the average end to end delay for clustering and non clustering scheme for different network sizes and this is shown in Figure 5 and Figure 6. We observed that the packet loss for network without clusters is more than the network with clusters and this is large for large network size due to the large volume of traffic generated.

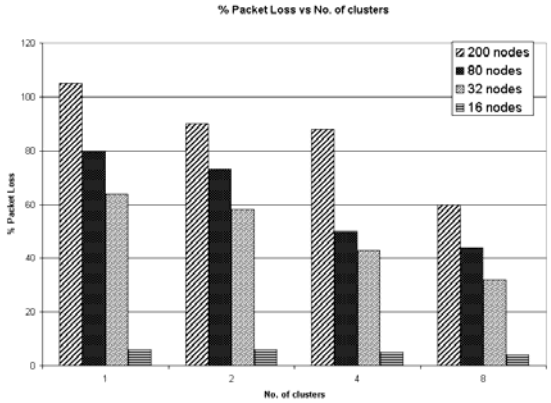

Fig.5 Graph showing the packet losses for varying network sizes as a function of the number of clusters

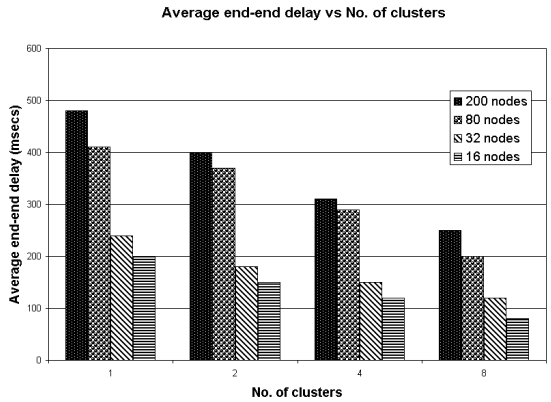

Fig.6 Graph showing the average end to end delay for varying network sizes as a function of the number of clusters

6 Conclusion

We proposed a decentralized approach of group key management that does not rely on a centralized authority for regenerating a new group key as well as for authentication. We also analyzed the performance of our approach in terms of security and found that it is robust against attacks and provides forward and backward secrecy. The results of the comparison of simulations of clustered and non clustered network are also presented.

References

[1] Rafaeli, S. and Hutchison, D. (2003) A survey of key management for secure group communication, ACM Computing Surveys, Vol. 35, No. 3, pp.309–329 .

[2] Burnett, S. and Paine, S. (2001). RSA Security's Official Guide to Cryptography, RSA Press

[3] Sherman, A.T. and McGrew, D.A. (2003) Key establishment in large dynamic groups using one-way function trees, IEEE Transactions on Software Engineering, Vol. 29, No. 5, pp.444–458

[4] Kim, Y., Perrig, A. and Tsudik, G. (2000) Simple and fault-tolerant key agreement for dynamic collaborative groups, Proc. 7th ACM Conference on Computer and Communications Security, ACM Press, pp.235–244.

[5] Bing Wu, Jie Wuand Yuhong Dong, (2008) An efficient group key management scheme for mobile ad hoc networks, Int. J. Security and Networks, 2008.

[6] MS. Bouassida, I. Chrisment, and O. Festor (2008). A Group Key Management in MANETs. in International Journal of Network Security, Vol.6, No. 1, PP.67-79, Jan. 2008

[7] Wenliang Du ,Ronghua Wang and Peng Ning (2005) An efficient scheme for authenticating public keys in sensor networks . MobiHoc'05, May 25–27, 2005, UrbanaChampaign, Illinois, USA

[8] The network simulator, http: // www.isi.nsnam/ns