Trust Based Hierarchical Key Management Scheme for Secure Group Communication in mobile Ad hoc networks

Mr. P. ANNADURAI
Department of Computer Science
(Government of Puducherry College)
KMCPGS, Lawspet Puducherry – 605 008 India Phone: 0413-2252128
annadurai_aps70@yahoo.co.in

Dr. V. PALANISAMY
Department of Computer Science & Engineering, Alagappa University
Karaikudi 630 003, Tamil Nadu, India,
Phone: 04565-226316
vpazhanisamy@yahoo.co.in

ABSTRACT

A mobile ad hoc network (MANET) is a kind of wireless communication infrastructure that does not have base stations or routers. Each node acts as a router and is responsible for dynamically discovering other nodes it can directly communicate with. However, when a message without encryption is sent out through a general tunnel, it may be maliciously attacked. The main idea in our proposal is key management for secure group communications in MANETs with two-layer structure. The level 1 subgroup (L1-subgroup) contains all of nodes in the subgroup. The level 2 subgroup (L2-subgroup) can be decided according to the location information of nodes in the L1-subgroup. Our scheme is to create a cluster head that manages information, and constructs and transmits the group key. First, in each subgroup, we select a node with the largest weight value to be the level 1 cluster head (L1-head) in each L1-subgroup. Then, in each L2-subgroup, the node with the largest weight value will be the level 2 cluster head (L2-head) and manage the other nodes of the L2-subgroup. With help of the cluster heads, the nodes authenticate each other and exchange their public key in a secure manner. The cluster head selection is based on the degree of node (i.e number of neighbors around the node) and node's identification number. Apart from these parameters, the member nodes asses trust of the cluster head. In this paper, we propose a trust based hierarchical key management scheme (TBHKMS) for secure group communications in MANETs. For the sake of security, we encrypt a packet twice. Due to the frequent changes of the topology of a MANET.

Keywords

Group communication; Group key; Key management; Mobile ad hoc networks; cluster head; Network security

1 INTRODUCTION

In an ad hoc network, there is a collection of mobile nodes that want to communicate to each others, but has no fixed links like wireless infrastructure network. Each node acts as a router and is responsible for dynamically discovering other nodes that it can directly communicate with. The emergence of the ad hoc networks raises a new challenge for the security of group communication, since the ad hoc networks are different from the traditional wired networks. So the security is a very important issue for wireless networks, especially for some security-sensitive applications.

The attributes of computer security – confidentiality, integrity, availability, authentication, and non-repudiation are valid for protection of the communications in ad hoc networks. Moreover, the network topology of an ad hoc network changes frequently and unpredictable, so routing and multicast become extremely challenging with the security issue.

In a MANET, a group can hasten message delivery and prevent bandwidth waste effectively. But if a message is sent out through a general tunnel without encryption, it may suffer malicious attacks. Because of these attacks, Internet security may be seriously affected. So in our scheme, a packet to be delivered will be encrypted, and only the receiver can decrypt the packet.

Key management schemes usually focus on improving security and reducing the memory storage of keys, as presented in MANETs. Two of the most common schemes for group structures are clustering and hierarchical trees. The advantage of clustering is that rekeying can be done quickly. The total cost of rekeying will increase greatly when members join or leave a larger group. Most group structures adopt a hierarchical tree. The main goal of a hierarchical tree is to decrease the cost of rekeying and to make management easy when changes in the group membership occur. The disadvantage of a hierarchical tree is that the maintenance cost increases when group membership increases [1].

Due to frequent changes of the network topology in a MANET, group maintenance of infrastructure wireless networks is not suitable. Therefore, we can use a common encryption key in a dynamic environment by following two rules. The first rule is forward secrecy. In this rule, when a new user joins a group, it cannot decrypt past encrypted messages. The second rule is backward secrecy. In this rule, when a group member leaves a group, it cannot decrypt future encrypted messages. If the two rules are followed, there will be better security for group key updating or protection. Managing keys efficiently within a group and reducing the amount of rekeying are the main goals we want to achieve. In this paper, we propose a trust based hierarchical key management scheme (TBHKMS) for secure group communications in MANETs. A secure group can manage members efficiently and reduce the amount of rekeying [3].

The rest of the paper is organized as follows. The improved model scheme (TBHKMS) is presented in Section 2. In Section3, we discuss the experimental results discussion. Finally, conclusions are given in Section 4.

2 IMPROVED MODEL SCHEME (TBHKMS)

In this section, we will introduce the key management concept and describe the group key maintenance in detail. First, Table 1 summarizes the notation used.

Table 1 Notation

Symbol	Meaning
$K_c(i,j)$	Communication key between two nodes i and j in different subgroups
K_{DH}	Private key that generated using DH
H_i	Level 1Trust cluster head in subgroup i
$H_{i,, j}$	Level 2 jth Trust cluster head in subgroup i
$L1GH_i$	Level 1 subgroup key in subgroup i
$L2GH_{i,,j}$	Level 2 jth subgroup key in subgroup i

2.1 Overview

The main idea in our proposal is key management for secure group communications in MANETs with two-layer structure. The level 1 subgroup (L1-subgroup) contains all of nodes in the subgroup. The level 2 subgroup (L2-subgroup) can be decided according to the location information of nodes in the L1-subgroup. Our scheme is to create a cluster head that manages information, and constructs and transmits the group key. Each cluster has a cluster-head and the other nodes of the cluster are member nodes. With the help of the cluster-heads, the nodes authenticate each other and exchange their public key in a secure manner. The cluster head selection is based on the degree of node and node's identification number. Apart from these parameters, the member nodes assess trust of the cluster head. First, in each subgroup, we select a node with the largest weight value to be the level 1 cluster head (L1-head) in each L1-subgroup. Then, in each L2- subgroup, the node with the largest weight value will be the level 2 cluster head (L2-head) and manage the other nodes of the L2-subgroup. Finally, one or several L2-heads will be obtained in the L1-subgroup. Figure 1 shows the subgroup key transmission operation between nodes. The procedures of the L1-head and L2-head selecting are described in Algorithm 1 and Algorithm 2, respectively [1].

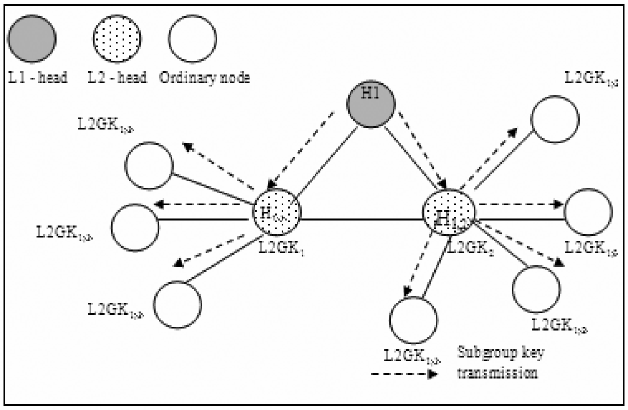

Figure 1: Subgroup key transmission operation between nodes.

Algorithm 1: L1-head selecting.

```
Step 1:  The weight value of each node with hello message is
         broadcasted to adjacent nodes. The delivery range of
         each node is not more than 2-hop.
Step 2:  After Step 1, we collect all weight values of nodes and
         select the largest one to be the L1-head.
Step 3:  check L1- head trust of node using Algorithm 3
          3.1  if node high trustable then set Head Elect = High
               Send HeadElect message to neighbors Wait for
               Head message.
          3.2  Else if node high not trustable then delete node
               high from array Node List (NLn) store high in
               blacklist array.
Step 4:  Other nodes will register to the selected L1-head and
         send all information to it.
```

Algorithm 2: L2-head selecting.

```
Step 1:  All nodes will send their location information to the
         L1-head.
Step 2:  After receiving the location information of all the nodes,
         according to the location information, the L1-head will
         classify all the nodes except the L1-head into      L2-
         subgroups.
Step 3:  The nodes of L2-subgroup will compare their weight
         values again and select the largest one to be the  L2-
         head.
Step 4:  check L2- head trust of node using Algorithm 3
         4.1 if node high trustable then set Head Elect = High
             Send HeadElect message to neighborsWait for Head
             message.
         4.2 Else if node high not trustable then delete node high
             from array Node List (NLn) store high in blacklist
             array.
Step 5:  The L2-head is to manage L2-subgroup and
         communicate with L1-head or other L2-subgroups.
```

2.2 Trust Establishment

As the cluster head plays a pivotal role in the existence of the cluster and later aid in the key management process. Naturally nodes will be interested to have this honor and can go to the very extent of providing false information about the neighborhood degree. This can lead to electing a node susceptible to provide incorrect information. So the responsibility of the Trust module is to ensure that node certified trust would be providing true information [3].

2.2.1 DURING THE HEAD ELECTION PROCESS

Nodes assess the trust of other nodes during election process when a Cluster Head Elect message is received or chooses a Cluster Head Elect or receives Head message. Under such events, nodes need to check the trust. The nodes retrieve the neighbor list for the node whose trust is to be evaluated. The common neighbors to both the nodes are left and nodes not in current node and present in neighbor list are the ones to be verified. So neighborhood status is enquired from those nodes. If at least half the responses are received then the nodes are trustable. Algorithm 3 discusses the trust evaluation procedures for a particular node.

Algorithm 3: Trust Evaluation for node

```
Step1:.  Array l = Get the neighbor list of node h from $NL_e$.
step2  :t =(the number of  nodes neighbor to h and not to e)/2
step3 :  for d Є nodes neighbor to h and not to e
        3.1 send message regarding the presence of h in d's
            neighbor list.
Step 4:  If the no. of positive response < t or Trust_interval
         timeout
        4.1 return false (node h cannot be trust)
Step5:   If the number of positive response > t
        5.1 return true.
```

2.3 Key Management Scheme

In the following, we will describe the subgroup key generation method and the packet delivery process in detail.

After the relationships among all subgroups are established, all nodes will send a registration packet to the L1-head. The L2-head knows the information of all the nodes in the L1-subgroup and generates the L1-subgroup key (L1GK) using RSA. After generating L1GK, the L1-head transmits it to all the nodes in the subgroup. L1GK is used to encrypt all the nodes in the subgroup[1].

In order to increase transmission security within the subgroup, we divide an L1-subgroup into L2-subgroups except the L1-head. Each L2-subgroup contains an L2-head and the nodes under the L2-head. These L2-subgroups will generate their own L2-subgroup keys (L2GKs). L2GKs are generated by calculating the known L1GK; hence, in a subgroup, in addition to the L2-heads and the nodes under them, other nodes will not get the L2GKs. For transmission between subgroups, we use DH to achieve transmission security. We first connect neighboring subgroups using the communication nodes in each subgroup. The communication key (Kc) is used for encryption and decryption of messages between two nodes in different subgroups. A communication node sends the information to the L1-head in order to generate communication keys which are used to connect subgroups. When an L2-head knows the location of the destination node, DH is also used to generate a private key KDH, which only belongs to the source node and the destination node. KDH will be used for the first encryption when the packet is transmitted. The packet will not be intercepted by the other nodes. Figure 2 shows the communications of mobile nodes in different subgroups. Each subgroup has its own subgroup key. Therefore, data delivery in different subgroups is only through subgroup keys and communication keys.

In figure 3, we assume that node A would like to send a packet to node D, and that the path of the destination node is known. First, source node A generates KDH by means of DH and encrypts the packet with KDH; this is the first encryption. Then L2GK1,1 is used to do the second encryption of the packet. The packet is then transmitted to L2-head H1,1. L2-head H1,1 will decrypt the packet with L2GK1,1, encrypt the decrypted packet with L1GK1, and then send the packet to the L2-head H1,2. After receiving the packet, L2-head H1,2 will decrypt the packet with L1GK1, encrypt the decrypted packet with L2GK1,2, and then send the packet to node B. After receiving the packet, node B will decrypt it with L2GK1,2, encrypt it with Kc, and then send it to node C. When the process of encryption and decryption is repeated, the packet will be transmitted to the destination node. After receiving the packet, destination node D will first decrypt it with L2GK2,1 and then decrypt it again with KDH to obtain the information in the packet.

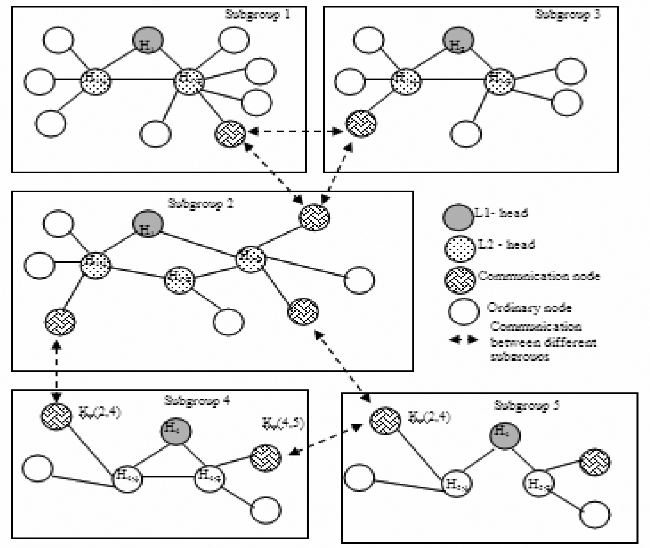

Figure 2: The node communication in different subgroups

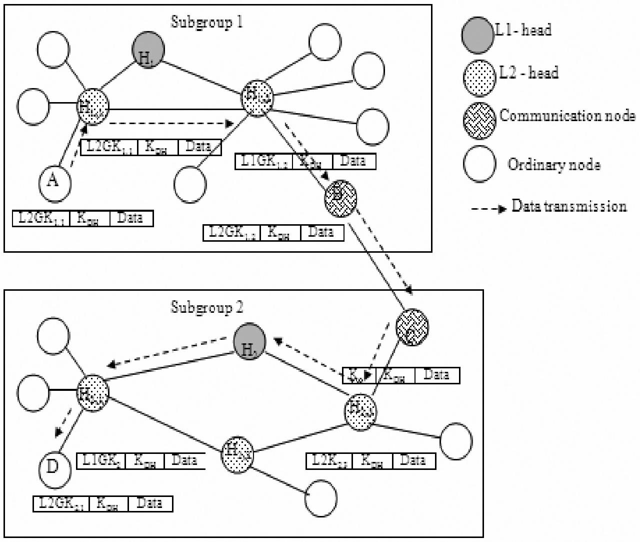
Subgroup 1
Subgroup 2
H_1
$H_{1,1}$
$H_{1,2}$
H_2
$H_{2,1}$
$H_{2,2}$
$H_{2,3}$
A
B
C
D
$L2GK_{1,1}$ | K_{DH} | Data
$L1GK_{1,2}$ | K_{DH} | Data
$L2GK_{1,2}$ | K_{DH} | Data
K_c | K_{DH} | Data
$L1GK_2$ | K_{DH} | Data
$L2K_{2,2}$ | K_{DH} | Data
$L2GK_{2,1}$ | K_{DH} | Data
L1- head
L2 - head
Communication node
Ordinary node
Data transmission

Figure 3: Encryption and decryption operation during data transmission.

2.4 Subgroup key maintenance

In the following, we will discuss the maintenance process required to the topological changes caused by nodes in a MANET.

2.4.1 NEW NODE JOINING A SUBGROUP

Each node in a MANET may not be fixed in one position and may sometimes moves frequently. We assume that every coming node is an authenticated node. Thus, we only regenerate a subgroup key after a new node joins a subgroup.

2.4.1.1 JOINING OPERATION

Group Join Request: The parameters of this request are group_id, TTL, and SN. The TTL value is dynamically set and, the sequence number SN is used to avoid multiple forwarding.

Receive Group Join Replies: Nodes that are already within the security multicast tree send replies to the requester. Unless if the number of connected nodes to them exceeds some threshold.

The replies contain information about: number of hops to the source, logical path to the source (the sequence of group members that lead to the source will be used in a handover to avoid loops), path quality to the source, and number of nodes already connected to this node. The requesting node will initiate a registration with the sender of the most satisfactory reply (in terms of aggregate path quality) [4, 5, 6 and 7].

Authentication, registration, and key establishment: The requester and intermediate node will first mutually authenticate each other. The authentication process will lead to the establishment of a shared key. Then, they will both check that they are allowed to access this information. This proof of access right is done using a service-access certificate.

Tree optimization: Once registered the newly joining node can send a path optimization message to nodes that are already in the tree but could optimize their path by attaching to the joining node. The joining node knows that from the request reply that they sent to him.

Receive encrypted data: The joining node can start receiving the encrypted data.

2.4.2 LEAVING OPERATION

Inform downstream nodes: The depending downstream nodes should initiate a handover and send a handover complete message once reconnected to the tree.

Inform upstream node: Once all downstream nodes are reconnected or after a timeout, the leaving node requests its upstream node to disconnect him from the tree.

3 EXPERIMENTAL RESULTS DISCUSSION

3.1 Metrics for evaluation

The metrics specific for the key management scheme based on grouping is the success rate of key agreement.

3.2 Success rate of key agreement

The simulation is run with 10% of total number of nodes as malicious nodes. Initially simulation is run without malicious nodes, then with malicious nodes and without trust model. Later with malicious nodes and with trust model. The success rate has been calibrated with number of successful agreed keys to number of request for key exchange.

1. CONCLUSIONS

It is very important to reduce bandwidth and protect the packet security during data transmission. In this paper we proposed a trust based hierarchical key management scheme (TBHKMS) for secure group communications in MANETs. For security, we protect our information from attacks by double encryption. We generate an L1-subgroup key for each L1-subgroup and an L2-subgroup key for each L2-subgroup. When the source node wants to send data to the destination node, they also will generate their own private key. The procedure of delivery is to encrypt the packet firstly by private key, and then encrypt and decrypt it again by L1-subgroup key and L2-subgroup key. Because, in a MANET, the topology changes frequently, the proposed TBHKMS can achieve the secure group communication in MANETs

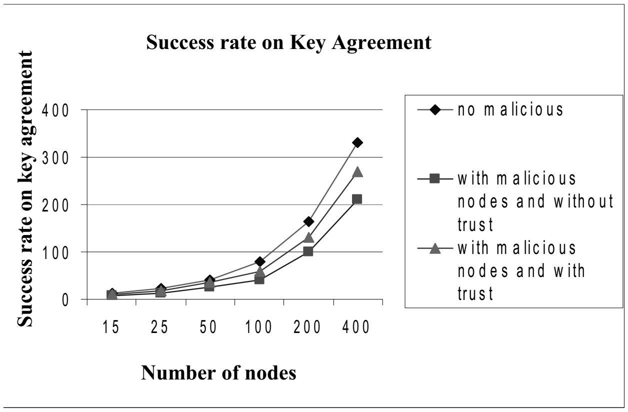

Figure 4: Success rate on key agreement

4 REFERENCES

[1] Nen-Chung Wang a,*, Shian-Zhang Fang, “A hierarchical key management scheme for secure group communications in mobile ad hoc networks”, Elsevier The Journal of Systems and Software 80 (2007) pages 1667–1677.

[2] Guangming Hu, Xiaohui Kuang, and Zhenghu Gong, “A Cluster-Based Group Rekeying Algorithm in Mobile Ad Hoc Networks”, Springer-Verlag Berlin Heidelberg 2005 ICCNMC 2005, LNCS 3619, pp. 344 – 353, 2005.

[3] T. Kaya, G. Lin, G. Noubir, A. Yilmaz , “Secure Multicast Groups on Ad Hoc Networks” Proceedings of the 1st ACM Workshop Security of Ad Hoc and Sensor Networks Fairfax, Virginia © 2003 ACM.

[4] Tzu-Chiang Chiang and Yueh-Min Huang “ Group Keys and the Multicast Security in Ad Hoc Networks “ Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPPW’03) 1530-2016/03 2003 IEEE.

[5] Ozkan M. Erdem, “ EDKM: Efficient Distributed Key Management for Mobile Ad Hoc Networks. Oregon State University, 0-7803-8623-W04/ IEEE.

[6] Ling Luo, Rei Safavi-Naini, Joonsang Baek and Willy Susilo, “Self-organised Group Key Management for Ad Hoc

Networks", ASIACCS '06 March 21 - 24, 2006, Taipei, Taiwan. Copyright 2006 ACM.

[7] Jun Li , Guohua Cui, Xiaoqing Fu, Zhiyuan Liu, Li Su, "A Secure Group Key Management Scheme in Mobile Ad Hoc Networks", Huazhong University of Science and Technology Wuhan , China, 2005 IEEE. \