A hierarchical key management scheme for secure group communications in mobile ad hoc networks

Nen-Chung Wang a,*, Shian-Zhang Fang b

a Department of Computer Science and Information Engineering, National United University, Miao-Li 360, Taiwan, ROC
b Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 413, Taiwan, ROC

Received 11 March 2006; received in revised form 24 November 2006; accepted 22 December 2006
Available online 23 January 2007

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. In this paper, we propose a hierarchical key management scheme (HKMS) 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, we also discuss group maintenance in this paper. Finally, we conducted the security and performance analysis to compare the proposed scheme with Tseng et al.’s [Tseng, Y.-M., Yang, C.-C., Liao, D.-R., 2007. A secure group communication protocol for ad hoc wireless networks. In: Advances in Wireless Ad Hoc and Sensor Networks and Mobile Computing. Book Series Signal and Communication Technology. Springer] and Steiner et al.’s [Steiner, M., Tsudik, G., Waidner, M., 1998. CLIQUES: a new approach to group key agreement. In: Proceedings of the 18th IEEE International Conference on Distributed Computing System. Amsterdam, Netherlands, pp. 380–387] schemes.

Keywords: Group communication; Group key; Key management; Mobile ad hoc networks; Network security

1. Introduction

A mobile ad hoc network (MANET) (Macker and Corson, 1998; Marwaha et al., 2002) is a kind of wireless communication infrastructure that does not have base stations or routers. Its investment cost is less, and each mobile node acts as a router on the Internet. A MANET can be deployed rapidly, and it can be used in remote places and battlefields, etc.

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 (Mirkovic et al., 2002; Mishra et al., 2004; Patwardhan et al., 2005; Russell, 2001; Schmooyer et al., 2004). 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 (Chang and Chung, 2003; Jablon, 1996; Rafaeli and Hutchison, 2003). Two of the most common schemes for group structures are clustering (Tseng et al., 2007) and hierarchical trees (Amir et al., 2004; Chiang and Huang, 2003; Liu and Zhou, 2002; Steiner et al., 1996; Wong et al., 2000; Yang and Zheng, 2001). 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.

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 hierarchical key management scheme (HKMS) for secure group communications in MANETs. A secure group can manage members efficiently and reduce the amount of rekeying.

The rest of the paper is organized as follows. In Section 2, we introduce the related work. The proposed scheme is presented in Section 3. In Section 4, we discuss the security analysis and time complexity analysis. In Section 5, we present the performance evaluation. Finally, conclusions are given in Section 6.

2. Related work

MANETs are typically dynamic peer networks (DPNs). The specific security requirements of DPNs (in particular, key management) are still considered to be open research challenges. Recently, several key agreement protocols for DPNs were proposed (Pieprzyk and Li, 2000; Steiner et al., 1996, 1998; Tseng et al., 2007; Yang and Zheng, 2001). In Steiner et al. (1996), the key agreement protocols were obtained by extending the well-known Diffie–Hellman (DH) key exchange scheme to groups of n parties. In Pieprzyk and Li (2000), two key agreement protocols were proposed based on the threshold cryptography using the Lagrange interpolation theorem.

A hierarchical structure is adapted by many key management schemes (Yang and Zheng, 2001). The basic rekeying algorithm of a hierarchical structure divides the key management domain into smaller administrative groups. In addition, public key infrastructure (PKI), secure multicast (Chan and Chan, 2003; Choudhary et al., 2004; Mukherjee et al., 2004), and logical tree-based algorithms are adapted by intra-group rekeying systems. Mobility impacts performance only when members cross groups. For instance, when two partners provide broadcast services for users in two overlapping groups, users moving within each group are managed by their local group key distributors (GKD) and without any coordination between their broadcasts. On the other hand, when a user crosses from one group to another, security should be transferred between partners.

Steiner et al. (1998) proposed a scheme called CLIQUES, which applies a key agreement rather than a key tree. CLIQUES is a logical linear structure which passes key information sequentially. The last group member will obtain the information of all the nodes. Tseng et al. (2007) proposed a safe communication scheme for MANETs. This communication is based on cluster structure and applies in wireless infrastructure. In this construction, the transmission range of nodes is 1-hop and one node with the largest weight value to be selected as multicast router (MR). The multicast router is a center to build the group. After group construction, MR will generate group key for encryption and decryption during data transmission. According to this scheme, we can manage keys efficiently and reduce the amount of rekeying.

3. The hierarchical key management scheme

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.

The topology changes frequently caused node’s moving in MANET. How to create and maintain one group is very important. We suppose the range which one node broadcast hello message to adjacent node is 2-hop. The hello message is to collect all information of nodes in the range of 2-hop. According to the information, we design the path and construct groups.

3.1. Key management

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 (Dhurandher and Singh, 2005; Huang et al., 2003; Pagani and Rossi, 1999; Purtoosi et al., 2004). 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{i,j}$</td>
<td>Communication key between two nodes $i$ and $j$ in different subgroups</td>
</tr>
<tr>
<td>$K_{DH}$</td>
<td>Private key that generated using DH</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Level 1 cluster head in subgroup $i$</td>
</tr>
<tr>
<td>$H_{i,j}$</td>
<td>Level 2 $j$th cluster head in subgroup $i$</td>
</tr>
<tr>
<td>$L1GK_i$</td>
<td>Level 1 subgroup key in subgroup $i$</td>
</tr>
<tr>
<td>$L2GK_{i,j}$</td>
<td>Level 2 $j$th subgroup key in subgroup $i$</td>
</tr>
</tbody>
</table>
will be obtained in the L1-subgroup. Fig. 1 shows the subgroup key transmission operation between nodes. The procedures of the L1-head and L2-head selecting are described in Procedure 1 and Procedure 2, respectively.

Procedure 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: Other nodes will register to the selected L1-head and send all information to it.

Procedure 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. The L2-head is to manage L2-subgroup and communicate with L1-head or other L2-subgroups.

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 (Pieprzyk and Li, 2000). 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.

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 (Ks) 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.

Fig. 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.

Fig. 3 shows the encryption and decryption operation during data transmission. In Fig. 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 encrypt it again with KDH to obtain the information in the packet.

Fig. 4 shows the data transmission procedure of the proposed scheme. Where the source node belongs is called the “subgroup 1”. We will call any other subgroup the “subgroup 2”. When a source node in subgroup 1 sends a packet to the destination node, it will double-encrypt the
Fig. 2. The node communications in different subgroups.

Fig. 3. Encryption and decryption operation during data transmission.
packet using $K_{DH}$ and L2GK$_{1,1}$ and transfer it to the L2-head 1 in subgroup 1. When the L2-head 1 in subgroup 1 receives this packet, it will decrypt the packet using L2GK$_{1,1}$ and encrypt it using L1GK$_1$. This packet is finally sent to the L2-head 2 in subgroup 2, and the decryption and encryption steps are repeated until the destination node receives this packet. When the destination node receives it, the packet will be decrypted completely using L2GK$_{2,1}$ and $K_{DH}$, and then the information of the packet can be obtained.

### 3.2. Subgroup key maintenance

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

Each node in a MANET may not be fixed in one position and may sometimes move 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.

Fig. 5 gives the scenario of a new node joining a subgroup. We assume that a mobile node C wants to join the subgroup. It will first send a join message to its neighboring nodes. When node A and node B receive a join message, they will send this message to their L2-head. When the L2-head receives the join message, the L2-head will respond with a message to the new node. For management and rekeying purposes, the L2-head also sends the new node’s information to the L1-head.

The detailed process of new node joining a subgroup is described below.

Step 1: A new node sends a join request message to the neighboring nodes.
Step 2: After receiving the join request message, the neighboring node sends it to the L2-head.
Step 3: The L2-head sends a reply message to the new node.
Step 4: The new node receives the reply message and then is allowed to join the L2-subgroup.
Step 5: The L2-head will regenerate an L2-subgroup key by updating the L2-subgroup information and send it to all L2-subgroup nodes.

3.2.2. Node leaving a subgroup

We will discuss three cases of different types of nodes leaving a subgroup: the leaving of ordinary nodes, the leaving of L2-heads, and the leaving of L1-heads.

Case 1: The leaving of ordinary nodes.

The leaving of ordinary nodes in a subgroup is simple. The L2-head only needs to regenerate a new L2GK. The keys of the remaining nodes will not be changed.

The detailed process of the leaving of ordinary nodes in a subgroup is described below.

Step 1: Before an ordinary node leaves the subgroup, it will send a leave message to the L2-head.
Step 2: After receiving the leave message, the L2-head will send a reply message to the leaving node and regenerate a new L2-subgroup key from the updated information of L2-subgroup. The new L2-subgroup key will send to all remaining nodes.

Case 2: The leaving of L2-heads.

We select an ordinary node with the largest weight value of the ordinary nodes in the subgroup to be the new L2-head. The remaining ordinary nodes will register to the new L2-head and send the information of all the nodes under them to the new L2-head.

Fig. 6 shows the example of an L2-head leaving a subgroup. When L2-head $L_{1,1}$ moves out of the subgroup, nodes under sub-head $L_{1,1}$ will select a node with the largest weight value to be the new L2-head. We assume that node $A$ is the new L2-head, and the remaining nodes will connect with node $A$. Node $A$ will forward new information to the L1-head. Then the L1-head will regenerate a
new L1GK, and transmit it to all L2-heads. The new L2-head will generate a new L2GK by means of L1GK and transmit the new L2GK to its ordinary nodes.

The detailed process of the leaving of L2-heads in a subgroup is described below.

Step 1: Before leaving the subgroup, the L2-head will send a leave message to the ordinary nodes and the L1-head. After receiving the leave message, the ordinary nodes and the L1-head will send a reply message to the leaving L2-head. If the L2-head does not receive the reply message in the period, it will send the leave message again to irresponsive nodes.

Step 2: After Step 1, the L2-subgroup managed by the leaved L2-head will select a new L2-head that contains the largest weight value by comparing the weight values of the ordinary nodes.

Step 3: The new L2-head will send the updated L2-subgroup information to the L1-head node.

Step 4: After receiving the updated information from the new L2-head, the L1-head will regenerate a new subgroup key and send it to all the L2-heads.

Step 5: After receiving the new subgroup key from the L1-head, the L2-head will regenerate a new L2-subgroup key and send it to all the ordinary nodes of L2-subgroup.

Case 3: The leaving of L1-heads.

We first select an L2-head with the largest weight value of the L2-heads in the subgroup to be the new L1-head. The remaining L2-heads will register to the new L1-head and send the information of all the nodes under them to the new L1-head. The nodes under the L2-head that has been selected to be the new L1-head will repeat the actions described in Case 2.

Fig. 7 shows the example of an L1-head leaving a subgroup. When the L1-head node leaves the subgroup, we will compare the weight values of L2-heads \( L_{1,1} \) and \( L_{1,2} \) to generate a new L1-head. If the weight value of \( L_{1,2} \) is larger than that of \( L_{1,1} \), it will be the new L1-head in the subgroup. Simultaneously, the previous L2-subgroup managed by \( L_{1,2} \) also will select a new L2-head from ordinary nodes by comparing their weight values. After reconstructing the subgroup, the new L1-head \( L_{1,2} \) will regenerate a new subgroup key, and then each L2-head will receive a new subgroup key from \( L_{1,2} \) and use it to generate a new L2-subgroup key.

The detailed process of the leaving of L1-heads in a subgroup is described below.

Step 1: Before leaving the subgroup, the L1-head will send a leave message to L2-heads. After receiving the leave message, the L2-heads will send a reply message to the L1-head. If the L1-head does not receive the reply message in the period, it will send the reply message again to the L2-heads.

Step 2: After receiving the leave message from the L1-head, the L2-heads of the same subgroup will compare their weight values and select the largest one to be the new L1-head node.

Step 3: Simultaneously, the ordinary nodes of L2-subgroup previously managed by the new L1-head compare their weight values and select the largest one to be the new L2-head node.

Step 4: All L2-heads of the subgroup have to send their L2-subgroup information to the new L1-head and register them.

Step 5: After receiving the information from the L2-heads, the new L1-head will regenerate a new subgroup key and send it to all L2-heads.

Step 6: After receiving the new subgroup key, the L2-head will regenerate a new L2-subgroup key and send it to the ordinary nodes of L2-subgroup.

In the ad hoc group, mobile nodes will move frequently, or even disappear suddenly. It causes that it is unable to maintain subgroup nodes, and sends data inefficiently. In order to prevent these situations, we discuss some specific cases:

Case 1: The leaving node disappears directly without sending the leaving message.

In our scheme, the L1-head sends the hello message in the fixed time to obtain the connections with the L2-heads and the L2-head also repeats the same process to obtain the connections with the ordinary nodes. If some nodes disappear, the subgroup will rearrange and generate a new subgroup key.
Case 2: Mobile nodes move frequently in every subgroup. Because the movement of mobile node will cause the subgroup to regenerate a subgroup key frequently. In order to prevent the situation, the generation of the new subgroup key will be held for a moment when the new node joins or leaves the subgroup.

4. Security analysis and time complexity analysis

Nodes in a subgroup are usually considered to be part of the security issue. It should be noted that nodes in a subgroup are secure because faster calculating speeds and various hacking methods may compromise their security. But, there are no fixed nodes to perform the service of authentication. We assume that all of the nodes of incoming subgroups in a MANET have already been acknowledged and have their own safety nodes with public and private keys. Therefore, the L1-head will generate L1GK and send it to all the L2-heads in the subgroup. Before the packet is sent, it will be encrypted with each public key of each node in the subgroup and then be sent to each node. When each node in the subgroup receives the packet, it will decrypt the packet with its own private key. Security is enhanced during transmission. In this section, we will analyze the security and performance analysis in detail.

4.1. Security analysis

In the following, we describe the security analysis of the proposed scheme.

(1) Forward secrecy.

We mention that new joining node could not receive the previous information before it joins. In the maintenance of our scheme, L2-head will regenerate subgroup key and transfer it to new node. New node just can encrypt and decrypt late information, but could not decrypt previous packet before it joins.

(2) Backward secrecy.

When a node leaves the subgroup, we have to make sure that node could not receive any information from the subgroup. We discuss the node leaving by the following three cases. Case 1 focuses on the leaving of ordinary nodes. A new L2-subgroup key is regenerated by the L2-head and sent to residual nodes in L2-subgroup. Case 2 discusses about the leaving of L2-head. In Case 2, the subgroup not only selects a new L2-head, but also regenerates subgroup key by the L1-head. The new subgroup key is sent to all L2-heads. Then each L2-head regenerates a new L2-subgroup key and sends it to ordinary nodes in L2-subgroup. Case 3 is for the leaving of L1-head. In this case, the new L1-head is selected from L2-heads of the subgroup and this new L1-head will repeat Case 2 to complete rekey. Following these three cases, we can make sure that the leaving nodes could not decrypt packets by previous subgroup key. Therefore, our scheme is fit to backward secrecy.

4.2. Time complexity analysis

In this subsection, we compare the proposed HKMS with Tseng’s and Steiner’s schemes as follows. We also give the summaries of time complexity comparison in Table 2.

We first define three parameters: (1) $m$ is the number of nodes in L2-subgroup; (2) $k$ is the branching factor of hierarchical structure or the number of L2-head; (3) $p$ is the subgroup size or the number of total nodes in the subgroup. We suppose that each L2-subgroup have the same number of members. Therefore, $p = mk + 1$. Then we give some comparisons in detail.

(1) Number of keys stored in an L1-head.

We use two-level construction to generate key. Beside one L1GK exists in the L1-head and the L2-head, there are $k$ L2-heads in the subgroup. After generating the L2-subgroup key (L2GK), the L2-head will send the L2GK to the L1-head. Therefore, all keys stored in the L1-head are $k + 1$ so its time complexity is $O(k)$. However, Steiner’s scheme uses Diffie Hellman scheme to generate key. In this scheme, two nodes must have one key and these keys finally combine together to generate their group key. For example, there are $p$ nodes in the group and total keys in the group are $p$. Therefore, this scheme has to store the most number of keys than others. On the other hand, Tseng’s scheme presents that there is only

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time complexity comparison</td>
</tr>
<tr>
<td>The proposed scheme (HKMS)</td>
</tr>
<tr>
<td>Number of keys stored in an L1-head</td>
</tr>
<tr>
<td>Number of keys stored in an L2-head</td>
</tr>
<tr>
<td>Number of keys stored in each member</td>
</tr>
<tr>
<td>Number of rekey cost in join process</td>
</tr>
<tr>
<td>Number of rekey cost in leave process</td>
</tr>
<tr>
<td>Number of retrieving a subgroup key</td>
</tr>
<tr>
<td>Communication inside subgroup</td>
</tr>
<tr>
<td>Vulnerable to implosion</td>
</tr>
</tbody>
</table>
one subgroup key for encryption and decryption in the subgroup. It causes key amount to be the fewest one of the three schemes.

(2) Number of keys stored in an L2-head.
In Steiner’s and Tseng’s schemes, they do not have any key stored in L2-heads. However, our HKMS provides two subgroup keys stored in L2-heads. One is L1-subgroup key (L1GK) for the connection between L1-head and L2-head. The other one is L2-subgroup key (L2GK) for the connection between L2-subgroups. In this term, the time complexity of the proposed HKMS is $O(1)$.

(3) Number of keys stored in each member.
The proposed HKMS and Tseng’s scheme provide only one subgroup key stored in each member. But in Steiner’s scheme, group key generation is to combine all keys of members. Therefore, the number of group keys of Steiner’s scheme is larger than the other two schemes.

(4) Node joining process.
The proposed HKMS mainly constructs on the hierarchical structure. When a new node wants to join the subgroup, we just regenerate L2-subgroup key (L2GK) of the L2-subgroup which members change and do not rekey whole subgroup. The time complexity of our scheme is $O(m)$. Nevertheless, in Steiner’s and Tseng’s schemes, when new node joins the subgroup, they have to regenerate new subgroup keys and send them to all members. The time complexity of the both schemes are $O(p)$.

(5) Node leaving process.
The node leaving process of the proposed HKMS is discussed by the following three cases. In Case 1, for the leaving of ordinary nodes, we just regenerate new L2-subgroup key (L2GK) for the L2-subgroup which members change, so the time complexity of the proposed HKMS is $O(m)$. In Case 2 and Case 3, when L1-head or L2-head leaves the subgroup, we have to regenerate subgroup key and send it to all the nodes, so the time complexity of the proposed HKMS is $O(p)$. However, in Steiner’s and Tseng’s schemes, the subgroup must regenerate new subgroup key and send it to all the nodes when node leaves the subgroup. The time complexity of the both schemes are $O(p)$.

(6) Processing time for retrieving a subgroup key.
Because the proposed HKMS use the L1-head to generate the L1-subgroup key (L1GK) and send it to the L2-heads to generate the L2-subgroup key (L2GK). Therefore, the time complexity of the processing time of the proposed HKMS is $O(1)$. But, in the other two schemes, the new subgroup key is directly sent to all members and the time complexity of the process time is $O(1)$.

(7) Communication inside a subgroup.
This term shows the time of encryption and decryption during data transmission. In Steiner’s and Tseng’s schemes, the packet is decrypted and encrypted only once. However, in the proposed HKMS, for data transmission in different L2-subgroups the packet is decrypted and encrypted twice, it costs longer transmission time than the other schemes.

(8) Vulnerable to implosion.
In Steiner’s scheme, group key generation has to pass several processes. It is more dangerous than the other two schemes in security.

5. Performance evaluation

5.1. Performance model

We develop a performance model based on the model developed in Chen et al. (2006) to evaluate the communication cost for secure group key management. The performance metric used in our simulation is based on the total communication cost per unit time incurred in response to secure group key management events including group join and group leave. Thus, the total communication consists of two components:

(1) Group join cost $C_{\text{join}}$: The cost for handling group join event. This cost also includes the cost caused by connection/disconnection events by group members.

(2) Group leave cost $C_{\text{leave}}$: The cost for handling group leave event. This cost also includes the cost caused by connection/disconnection events by group members.

As a result, the total communication cost $C_{\text{total}}$ is calculated by

$$C_{\text{total}} = C_{\text{join}} + C_{\text{leave}}$$

Before the detailed explanation of our performance metric model, we first give the basic model parameters in Table 3.

A group join event requires the update the group key and the rekeying of group key from which the join event is originated, the cost of which is $C_{\text{group}}$. Therefore,

$$C_{\text{join}} = \frac{N - N_{\text{leader-L1}} - N_{\text{leader-L2}}}{N} \times C_{\text{group}}$$

where $C_{\text{group}}$ is given by

$$C_{\text{group}} = C_{\text{update}} + C_{\text{rekey}}$$

with

$$C_{\text{update}} = H_{\text{leader}} \times M_{\text{leader}} + M_{\text{member}} \times N_{\text{member}} \times M_{\text{update}}$$

and

$$C_{\text{rekey}} = H_{\text{region}} \times N_{\text{member}} \times M_{\text{region-member}}$$
The cost for group leave event includes three cases: non-leader member leaves, L1-leader leaves, and L2-leader leaves. Thus, the cost for group leave event is

\[
C_{\text{leave}} = C_{\text{non-leader}} + C_{\text{leader-L1}} + C_{\text{leader-L2}}
\]

with

\[
C_{\text{non-leader}} = \frac{N - N_{\text{leader-L1}} - N_{\text{leader-L2}}}{N} \times C_{\text{group}}
\]

\[
C_{\text{leader-L1}} = P_{\text{leader-L1}} \times (C_{\text{group}} + C_{\text{change-leader-L1}})
\]

\[
C_{\text{leader-L2}} = P_{\text{leader-L2}} \times C_{\text{change-leader-L2}}
\]

\[
P_{\text{leader-L1}} = \frac{N_{\text{L1}} / N_{\text{L1}}}{N_{\text{L1}}} + \frac{N_{\text{L1}}}{N_{\text{member}}} + \frac{N_{\text{L2}}}{N_{\text{member}}}
\]

\[
P_{\text{leader-L2}} = \frac{N_{\text{L1}} / N_{\text{L1}}}{N_{\text{L1}}} + \frac{N_{\text{L1}}}{N_{\text{member}}} + \frac{N_{\text{L2}}}{N_{\text{member}}}
\]

where \(C_{\text{group}}\) is given earlier in Eq. (3).

5.2. Performance analysis

We compared the performance of the proposed scheme with that of Tseng’s scheme and Steiner’s scheme. Nodes are deployed in a 1000 m x 1000 m square area. The radio radius is 100 m. Simulation time is 30 s of 60 times.

Fig. 8 shows that the effect of number of nodes on the overall cost with the proposed scheme, Tseng’s scheme, and Steiner’s Scheme. The cost of our proposed scheme is fewer than the cost of Tseng’s scheme and Steiner’s scheme. Note that the overall cost of Steiner’s scheme is the same as Tseng’s scheme.

Overall cost vs. mobility speed is compared in Fig. 9. There are 250 nodes deployed in a 1000 m x 1000 m square area. Our proposed scheme can reduce the cost under different speed of the nodes.

6. Conclusions

It is very important to reduce bandwidth and protect the packet security during data transmission. In this paper, we proposed a hierarchical key management scheme (HKMS) 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 HKMS can reduce the cost of rekey procedure. Finally, the security and performance analysis were conducted to compare the proposed scheme with Tseng et al.’s (2007) and Steiner et al.’s (1998) schemes.

Acknowledgements

This work was supported by the National Science Council of Republic of China under grants NSC-94-2213-E-324-025 and NSC-95-2221-E-239-052.
References


Nen-Chung Wang received the B.S. degree in Information and Computer Engineering from Chung Yuan Christian University, Taiwan, in June 1990, and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Cheng Kung University, Taiwan, in June 1998 and June 2002, respectively. From 2002 to 2006, he was an Assistant Professor in the Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taiwan. Since August 2006, he joined the faculty of the Department of Computer Science and Information Engineering, National United University, Taiwan. He is a member of the Phi Tau Phi Society. His current research interests include computer networks, wireless networks, parallel and distributed computing, e-learning, and bioinformatics. He is a member of the IEEE Computer Society.

Shian-Zhang Fang received the B.S. degree in Communication and Computer from Da Yeh University, Taiwan, in June 2003, and the M.S. degree in Computer Science and Information Engineering from Chaoyang University of Technology, Taiwan, in June 2005. His research interests include wireless networks, mobile computing, and wireless security.