

Making the Key Agreement Protocol in Mobile Ad Hoc Network More Efficient

Gang Yao^{1,2}, Kui Ren¹, Feng Bao¹, Robert H. Deng¹, and Dengguo Feng²

¹ InfoComm Security Department, Institute of InfoComm Research,
21 Heng Mui Keng Terrace, Singapore 119613

{yaogang, renkui, baofeng, deng}@i2r.a-star.edu.sg

² The State Key Laboratory Of Information Security, Institute of Software,
Chinese Academy of Sciences, Beijing 100080, P.R.China
fengdg@263.net

Abstract. Mobile ad hoc networks offer convenient infrastructureless communications over the shared wireless channel. However, the nature of mobile ad hoc networks makes them vulnerable to security attacks, such as passive eavesdropping over the wireless channel and denial of service attacks by malicious nodes. To ensure the security, several cryptography protocols are implemented. Due to the resource scarcity in mobile ad hoc networks, the protocols must be communication efficient and need as less computational power as possible. Broadcast communication is an important operation for many application in mobile ad hoc networks. To securely broadcast a message, all the members in the network need share a group key so that they can use efficient symmetric encryption, such as DES and AES. Several group key management protocols have been proposed. However, not all of them are communication efficient when applied to mobile ad hoc networks. In this paper, we propose a hierarchical key agreement protocol that is communication efficient to mobile ad hoc networks. We also show how to manage the group efficiently in a mobile environment.

Keyword: Mobile ad hoc networks, key agreement, hierarchical, security.

1 Introduction

An ad hoc network is a collection of nodes dynamically forming a peer to peer network without the use of any existing network infrastructure or centralized administration. Mobile ad hoc networks have attracted significant attentions recently due to its wide applications in different areas. Nodes of a mobile ad hoc network are often mobile, and the network can be formed, merged together or partitioned into separate network on the fly, without necessary relying on a fix infrastructure to manage the operation. Mobile nodes that are within the communication range of each other can communicate directly, whereas the nodes that are far apart have to rely on intermediary nodes (routers) to relay messages. Mobile ad hoc networks can be used for emergency, law enforcement and rescue

missions. Since, the cost to set up a mobile ad hoc network is low, it is also a very attractive option for commercial uses.

Security is one of the most important implementation issues for mobile ad hoc networks and it must be solved. There are four fundamental security issues which must be addressed: confidentiality, authentication, integrity and availability. Because of the high level of self-organization, dynamic topology, dynamic membership or vulnerable wireless link, mobile ad hoc networks are difficult to secure. In addition, security solutions applied in more traditional networks may not directly be suitable for protecting them.

Because the mobility of a node in the mobile ad hoc network causes frequently changed network topology, any security solution with a static configuration is not suitable. Till now, almost all cryptography protocols are based on secret keys or public keys. Public key based protocols have some inherent advantages over the secret key algorithms. However, it is well-known that the secret key based encryption algorithms (such as DES, AES) is much faster than the public key based protocols (such as RSA, ElGamal). In this paper, we will concentrate on how to build a common secret key for a group so that they can communicate securely. Several group key management protocols ([1], [2], [11], [12], [17], [18]) have been proposed for wired networks. However, not all of them are communication efficient when applied to mobile ad hoc networks. In this paper, we propose a hierarchical key agreement protocol that is communication efficient by using hierarchical networks to set up clusters among a group of mobile nodes. We apply some existed group key agreement protocols for each cluster at each level. As long as the selected protocols used in our protocol are secure, our protocol is also secure unambiguously.

The rest of the paper is organized as follows. In section 2, we recall some basic definitions and review several previous key agreement protocols. In section 3, we present our hierarchical key agreement protocol and give the communication cost, lasting time and total exponentiation of the protocol. In section 4, we show how to dynamic maintain the members in our protocol. And the conclusion is given in section 5.

2 Preliminaries

In this section, we present some definitions and general terminologies used in this paper, and we also review some related work.

2.1 Basic Definitions

The definitions have been adapted from [2], [18] and [15]. In general, the key establishment protocols can be classified into two types: key distribution protocols and key agreement protocols. The key distribution protocols, sometimes called as *centralized* key distribution protocols, are generally based on a trusted third party (TTP). The key agreement protocols, on the other hand, do not use a TTP but rely on the group members for a general key agreement. A *key agreement*

protocol is a key establishment method in which, a shared secret key is derived by two or more specified parties as a function of information contributed by, or associated with, each of these, such that no party can predetermine the resulting value.

Let \mathcal{P} be a n -party key agreement protocol, and \mathcal{M} be the set of members in the protocol and let S_n be a secret key generated as a result of \mathcal{P} . The protocol \mathcal{P} is said to provide *implicit key authentication* if each $M_i \in \mathcal{M}$ is assured that no party $Q \notin \mathcal{M}$ can learn the key S_n . Then this protocol is called an *authenticated key agreement*. The protocol \mathcal{P} provides *key confirmation* if any member is assured that its peer(s) do in fact, possess the particular key S_n . A key agreement protocol offers *perfect forward secrecy* if the compromising of a long-term key S_n cannot result in the compromising of the keys generated before S_n . On the other hand, a key agreement protocol is said to be vulnerable to *known key attacks* if the compromising of past keys allows a passive adversary to get future group keys, or an active adversary to impersonate one of the protocol members. For more detailed discussions of the above definitions, see [2], [4] and [5].

Then we present some of the definitions used in a mobile ad hoc environment. Assume that all mobile nodes are given as a set V of n nodes. Each node is assumed to have some computational power and an omni-directional antenna. A message sent by a node can be received by all nodes within its transmission range. Here we assume every node has the same maximum transmission range which is normalized to one unit. All these nodes induce a *unit disk graph* $UDG(V)$, in which, there is an edge between two nodes if and only if the distance between them is at most one unit. The $UDG(V)$ is always assumed to be a connected graph. All the nodes within a constant k -hop neighborhood of a node $u \in V$ are the k -local nodes or k -hop neighbors of u , represented by $N_k(u)$ hereafter. All nodes are assumed to be almost static for a reasonable period of time.

A type of hierarchical ad hoc network structure is called an *ad hoc network with mobile backbones* (MBN). In the mobile ad hoc networks, mobile nodes are first dynamically grouped into 1-hop clusters. Each cluster elects a cluster head to be a *backbone node* (BN). Among the mobile nodes, backbone nodes have an additional powerful radio to establish wireless links among themselves. Thus, higher level links are established to connect the BNs into a network, and we call this higher level network a *backbone network*. Since the backbone nodes are also moving and join or leave the backbone network dynamically, the backbone network is exactly an ad hoc network running in a different radio level. Multilevel MBNs can be formed recursively in the same way. For more detailed discussions of the above definitions, see [10], [16] and [21].

2.2 Some Secure Group Key Agreements

Before we review the group key agreement protocols, we describe the notations used in these protocols:

n	number of members in the protocol
i, j, k	indices of members (range $[1, n]$)
M_i	i -th group member
q	order of an algebraic group G
α	exponential base delimited by q
r_i	random exponent generated by M_i
K	Group key shared among n members

The Diffie-Hellman algorithm [8] allows the establishment of a cryptographic secret key between two entities by means of data exchange through an insecure communication channel. The algorithm executed between two entities M_1 and M_2 is defined as follows: Member M_1 sends α^{r_1} to M_2 and M_2 sends α^{r_2} to M_1 . M_1 computes the key $K = (\alpha^{r_2})^{r_1}$ and vice-versa for M_2 . The security of this protocol is based on the assumption of the difficulty of the discrete logarithm arithmetic and the Diffie-Hellman Decision problem.

Several solutions for extending the Diffie-Hellman key exchange to a multiparty key agreement have been proposed, such as the group Diffie-Hellman protocol [17], the hypercube protocol [7], the octopus protocol [7], the Burmester-Desmedt protocol [6], the tree based protocol [6] and etc.

The generic n -party Diffie-Hellman key exchange protocols were developed by Steiner, Tsudik and Waidner [17]. These protocols suite consists of key management protocols for dynamic groups. Two protocols GDH.2 and GDH.3 were presented.

Algorithm 1: The group Diffie-Hellman protocol GDH.2

- Round i ($1 \leq i \leq n - 1$): M_i sends $\alpha^{\prod_{k=1}^i r_k}$ and $\alpha^{(\prod_{k=1}^i r_k)/r_j}$ ($\forall 1 \leq j \leq i$) to M_{i+1} .
- Round n : M_n sends $\alpha^{(\prod_{k=1}^n r_k)/r_i}$ to each M_i .

Each member computes the final key as $K = (\alpha^{(\prod_{k=1}^n r_k)/r_i})^{r_i} = \alpha^{(\prod_{k=1}^n r_k)}$.

Algorithm 2: The group Diffie-Hellman protocol GDH.3

- Round i ($1 \leq i \leq n - 2$): M_i sends $\alpha^{\prod_{k=1}^i r_k}$ to M_{i+1} .
- Round $n - 1$: M_{n-1} sends $\alpha^{\prod_{k=1}^{n-1} r_k}$ to each M_i .
- Round n : M_i sends $\alpha^{(\prod_{k=1}^{n-1} r_k)/r_i}$ to M_n .
- Round $n + 1$: M_n sends $\alpha^{(\prod_{k=1}^n r_k)/r_i}$ to M_i .

Each member computes the final key as $K = (\alpha^{(\prod_{k=1}^n r_k)/r_i})^{r_i} = \alpha^{(\prod_{k=1}^n r_k)}$.

The hypercube protocol was presented by Becker and Wille [7] as an example of the protocol requiring the minimum number of rounds. Four nodes, which we shall call M_1, M_2, M_3 and M_4 , are arranged in a square. They can create a shared key by just four Diffie-Hellman key exchanges:

Algorithm 3: The hypercube protocol ($n = 4$)

- Round 1: M_1 and M_2 exchange keys ($\alpha^{r_1 r_2}$) in the usual way, and M_3 and M_4 exchange keys ($\alpha^{r_3 r_4}$) in the usual way.
- Round 2: Next, M_1 exchanges keys with M_3 , using the 2-way key as the secret exponent, and M_2 and M_4 do the same.

The result $K = \alpha^{\alpha^{r_1 r_2} \alpha^{r_3 r_4}}$ is a key that is shared among all four participants. This is the hypercube protocol in the case where the dimension of the cube d equals 2. The 4-way key exchange can be generalized to a protocol for higher numbers of nodes, as long as the number of participants equals 2^d , for $d \in \mathbb{Z}$, and the protocol executes in d rounds.

The octopus protocol is an extension of the hypercube protocol for networks with an arbitrary number of nodes. A subgroup of nodes is arranged in a hypercube, composing a core. Each core node establishes a key with each nearby non-core node using the Diffie-Hellman protocol. The product of these keys is used to establish a key among the core nodes as specified by the hypercube protocol. At last, this key is distributed to the other nodes.

In [6], Burmester and Desmedt presented several group key distribution systems based on public keys. They also extend these protocols to authentication and prove the security provided the Diffie-Hellman problem is intractable. From the ad hoc network point of view, the most interesting protocol is the tree based protocol. This is a key distribution protocol for a network whose topology is in the form of a binary tree. The root is a chair that generates the key and distributes it along the tree. The participants' contributions are only used in encrypting the session key while distributing it.

Algorithm 4: The tree based key distribution protocol

- Round 1: M_i computes $z_i = \alpha^{r_i}$. If $i > 1$, M_i sends z_i to $M_{\lfloor i/2 \rfloor}$; If $2i \leq n$, M_i sends z_i to M_{2i} ; If $2i + 1 \leq n$, M_i sends z_i to M_{2i+1} .
- Round 2: M_i computes $K_i = z_{\lfloor i/2 \rfloor}^{r_i}$ if $i > 1$, and $K_{2i+j} = z_{2i+j}^{r_i}$ for $j = 0, 1$, if $2i + j \leq n$.
- Round 3: M_1 selects a session key K , then he sends $Y_{2+j} = K \cdot K_{2+j}$ to M_{2+j} for $j = 0, 1$, and set $l = 0$.
- Round 4 + l : If M_i is at level l of the tree ($\lfloor \log_2 i \rfloor = l$), then M_i decrypts Y_i in order to get K . Next, he sends $Y_{2i+j} = K \cdot K_{2i+j}$ to M_{2i+j} for $j = 0, 1$, if $2i + j \leq n$, and set $l = l + 1$.

Burmester and Desmedt [6] were presented another key agreement protocol which is executed in three rounds. Each participant M_i ($i \in [1, n]$) executes the following operations:

Algorithm 5: The Burmester-Desmedt protocol

- Round 1: M_i generates a secret random value r_i and broadcasts $z_i = \alpha^{r_i}$ to the other participants;
- Round 2: M_i computes and broadcasts $X_i = (\frac{z_{i+1}}{z_{i-1}})^{r_i}$ to the other participants;
- Round 3: M_i computes the group key $K = z_{i-1}^{nr_i} \cdot X_i^{n-1} \cdot x_{i+1}^{n-2} \cdot \dots \cdot X_{i-2} \pmod p$.

This group key has the form $K = \alpha^{r_1 r_2 + r_2 r_3 + \dots + r_n r_1}$ and shares the security characteristics presented by the Diffie-Hellman algorithm. This protocol is

efficient with respect to the total number of rounds. This characteristic could allow faster execution, but each round requires n simultaneous broadcasts which are usually not possible, even in wireless networks. Another disadvantage is that this protocol makes use of a high number of exponential operations.

At last, we review the hybrid key agreement protocol presented by Li, wang and Frieder [14]. In this protocol, the first round is a clustering method that divides the entire set of the nodes into subgroups based on the geometric locations of the nodes. Each of these subgroups selects a leader (also called dominator). This selection process is done by generating a connected dominating set (CDS) (see [19], [3] and [20]) from the set of wireless nodes. Once the CDS has been constructed, the set of dominators of the CDS form the subgroup leaders; each dominator and the set of its dominatees form an individual subgroup. Then the key agreement protocols such as GDH.2 etc. could be applied to the set of dominators, and the key is generated as a contribution from all the dominators. On success of the key agreement protocol over the dominators, each dominator and the set of its subgroup members follow the key distribution protocol if the same key for each subgroup is required. The afore mentioned steps make sure that all the nodes share the same key, and an overall key agreement is reached.

Algorithm 6: The hybrid key agreement protocol

1. Wireless nodes construct a CDS distributively.
2. The contributory key agreement protocol is applied among the set of computed dominators and connectors.
3. Each dominator distributes the computed key to all its dominatees if the same key is required. Otherwise, each subgroup performs its own key agreement protocol.

Because this protocol just establish one group key and maintain the topology of the network while the node is moving in the network, it is not very efficient for using in the mobile ad hoc network.

3 Hierarchical Key Agreement Protocol

In a mobile ad hoc environment, the number of members could be very large. If all the members participate the process that creates the common secret key, it will be very difficult to manage the process. A method to solve this problem is to build a hierarchical ad hoc network. In each level, all the members are divided into clusters and all the members in a cluster perform a key agreement protocol to get the cluster key. Then all the cluster leaders perform a key agreement protocol to get group key. At last, the group key is distributed to all the group members.

3.1 Protocol

We present a hierarchical key agreement protocol in this subsection. Our protocol is a hierarchical link state based key agreement protocol. When the group key is

being established, the mobile ad hoc network maintains a hierarchical topology, where elected cluster heads at the lowest level become the members of the next higher level. These new members in turn organize themselves in clusters, and so on. The goal of clustering is to establish cluster key with high efficiency. Once the hierarchical structure of the network has been constructed, we could apply the key agreement protocol to establish the group key. On success of the key agreement protocol over each level, each cluster leader and the set of its cluster members follow the key distribution protocol so that the upper level keys can be distributed to the members of this lower level. The afore mentioned steps make sure that all the nodes in the mobile ad hoc network share the same group key, and an overall key agreement is reached. We outline our hierarchical key agreement protocol that is suitable for mobile ad hoc networks as follows:

Algorithm 7: The hierarchical key agreement protocol

- Step 1: Mobile nodes in the ad hoc network construct a hierarchical link networks of h levels. The clusters are independently controlled and are dynamically reconfigured as nodes move.
- Step $i + 1$ ($1 \leq i \leq h$): Each cluster in level i :
 1. Each member u random chooses a secret r_u .
 2. All the members agree with a key agreement protocol.
 3. All the members perform the protocol to establish the cluster key.
- Step $h + 2$: Each cluster leader distributes the computed upper cluster keys to all the member in the cluster.

For example, suppose that the hierarchical ad hoc network has three levels after all members execute the first step. We denote the cluster in level i by $C_{i,j}$ ($i = 1, 2, 3$), then at the end of step 2, the members in cluster $C_{i,j}$ will share the cluster key $K_{i,j}$. At the third step, each cluster leader distributes the upper cluster keys to all the member in the cluster. In the figure, the group key will be $K_{3,1}$. All the members in cluster $C_{2,1}$ will share the cluster key $K_{2,1}$ and the group key $K_{3,1}$, and all the members in cluster $C_{1,1}$ will share the cluster key $K_{1,1}$, higher cluster key $K_{2,1}$ and the group key $K_{3,1}$. All the members in the mobile ad hoc network can exchange the message encrypted by $K_{3,1}$; the member in the cluster $C_{1,1}$, $C_{1,2}$ and $C_{1,3}$ can exchange the message encrypted by $K_{2,1}$; the member in the cluster $C_{1,1}$ can exchange the message encrypted by $K_{1,1}$.

In addition to multilevel clustering, our protocol also provides multilevel logical partitioning. While clustering in the key establishment phase is based on geographical relationship between the mobile nodes, logical partitioning is based on logical, functional affinity between the mobile nodes. After the group key is established, we will only maintain the logical topology of the clusters, not the geographical topology. When the member in the network moves, if the logical topology of the network does not change, we will not change the group key, although the geographical topology may has been changed.

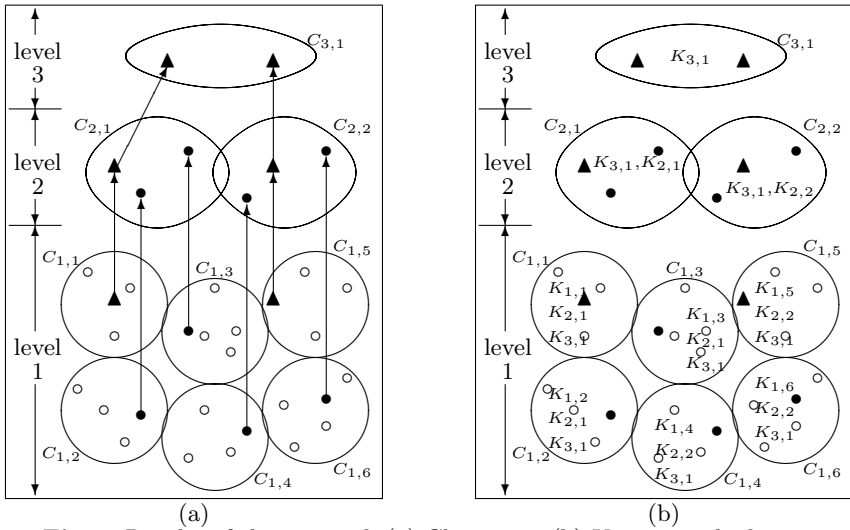


Fig. 1. Results of the protocol: (a) Clustering. (b) Keys in each cluster.

3.2 Analysis

In this subsection, we concentrate on the analysis of the communication complexity, computation complexity and time complexity of our hierarchical key agreement protocol. To simplify the analysis, we assume that the hierarchical ad hoc network has only two levels.

Most key agreement algorithms assume that an existing order is defined in the group members by requiring each member M_i sending a message to M_{i+1} , and the communication cost between any two members M_i and M_{i+1} is always one unit. In practice, these assumptions do not hold in wireless ad hoc environment. The nodes on the backbone are not connected to each other in any specific order, and a direct communication exists only between a node M_i and its 1-hop neighbors. If M_{i+1} is not the 1-hop neighbor of M_i , then the communication from M_i to M_{i+1} must be relayed by other intermediate nodes.

Assume that there are total n wireless nodes. Different clustering algorithms, such as Lowest ID and Highest Degree algorithm (more detail, see [9] and [13]) can be used for the creation of the clusters and the election of cluster header. After the clustering process, there are g clusters C_1, C_2, \dots, C_g with n_1, n_2, \dots, n_g members. The members in cluster C_i are denoted by $M_{i,j}$ ($i = 1, 2, \dots, g, j = 1, 2, \dots, n_i$), respectively, and the cluster leader in C_i is denoted by D_i . All the D_i form the backbone network. The communication cost of the clustering step of algorithm 7 is $O(n)$.

In the following discussion, we assume in each cluster, the communication cost by sending a message from one member to the other is one unit, so we can easily get the efficiency parameters for each protocol in any cluster.

	GDH.2	GDH.3	Hypercube	B-D	Tree Based
comm. cost	n_i	$2n_i - 1$	$n_i \log n_i$	$2n_i$	$2n_i - 1$
lasting time	n_i	$n_i + 1$	$\log n_i$	2	$\log n_i + 1$
total exp.	$n_i(n_i + 3)/2 - 1$	$5n_i - 6$	$2n_i \log n_i$	$n_i(n_i + 1)$	$2n_i$

Next, we establish the efficiency parameters for the backbone network. Assume that the average communication cost between two neighbor in the backbone network is c_1 and the broadcasting communication cost in the backbone network is c_2 . Assume that the average time for transmitting a message between two neighbor in the backbone network is t_1 and the longest time for transmitting a message between two nodes in the backbone network is t_2 . Then, we can easily get the efficiency parameters for each protocol in the backbone network. For the tree based protocol is a key distribution protocol, we do not compare it with the others.

	GDH.2	GDH.3	Hypercube	B-D	Tree Based
comm. cost	$(g - 1)c_1 + c_2$	$(g - 2)c_1 + 3c_2$	$(g \log g)c_1$	$2gc_2$	$gc_1 + c_2$
lasting time	$(g - 1)t_1 + t_2$	$(g - 2)t_1 + 3t_2$	$t_1 \log g$	$2t_2$	$t_1 + t_2$
total exp.	$g(g + 3)/2 - 1$	$5g - 6$	$2g \log g$	$g(g - 1)$	$2g$

It is difficult to establish the minimal communication cost and lasting time in the mobile ad hoc network, for it is dependent on the geographical topology of the network. In the following, we discuss the minimal communication cost and lasting time when the geographical topology of the mobile ad hoc network is stable.

It is easy to see that the total communication cost of this protocol is

$$\sum_i (\text{cost of cluster } C_i) + (\text{cost of backbone}).$$

In our protocol, using GDH.2 in clusters and the backbone network can get the minimal communication cost, and it is n_i in the cluster C_i and $(g - 1)c_1 + c_2$ in the backbone network. Thus, the minimal total communication cost of the hierarchical key agreement protocol is $\sum_i n_i + gc_1 + c_2 = n + gc_1 + c_2$.

Because the cluster key agreement can execute paralleled, the total lasting time of the hierarchical key agreement protocol is

$$\max_i \{\text{time of cluster } C_i\} + (\text{time of backbone}).$$

If the broadcasting operation is forbidden, the minimal lasting time can be got by using the hypercube protocol in clusters and the backbone network, and the lasting time of the hierarchical key agreement protocol is $\max_i \{\log n_i\} + t_1 \log g$. If the broadcasting operation can be used, the minimal lasting time can be got by using the Burmester-Desmedt protocol in clusters and the backbone network, and the lasting time of the hierarchical key agreement protocol is $2 + 2c_2$.

The total exponential operations of this protocol is

$$\sum_i (\text{exp. operations of cluster } C_i) + (\text{exp. operations of backbone}).$$

In our protocol, using GDH.3 in clusters and the backbone network can get the minimal total exponential operations, and the total exponential operations of the hierarchical key agreement protocol is $\sum_i (5n_i - 6) + (5g - 6) = 5n - g - 6$.

The last step of the hierarchical key agreement protocol is to distribute the group key to all the group members. In this step, each cluster leader broadcasts a message to all the members in the cluster. Because we assume that all the members in a cluster are in 1-hop area, the total message in each cluster is one. Thus, the communication cost of this step is $O(g)$.

4 Dynamic Maintenance

In the previous protocols, they assumed that the nodes in the network are static or can be viewed static. This is not true in mobile ad hoc networks. Mobile nodes will move around in the network. The movement of the nodes makes it very difficult to design efficient protocols for various applications such as routing, backbone construction, and so on. The hybrid key agreement [14] consider the movement of the nodes, but this protocol maintain the geographical topology of the nodes, and the group key will update when a node moves into or out of a subgroup. If the nodes in the network moves frequency, it will take lots of communication and time to manage the group key, and this makes the hybrid key agreement not very efficient for the mobile ad hoc network. In this section, we study in detail how our hierarchical key agreement protocol can easily manage the group key in the network. If the node's movement does not cause the change of the logical topology of the network, it is obvious that no key updating is necessary, although the geographical topology of the network maybe changes. We detail our hierarchical key agreement how to dynamic manage the group key in the mobile as hoc network. We assume that GDH protocol is used in both the clusters and the backbone network.

4.1 Member Joining

We first show that how a new member u join in the mobile ad hoc network.

Suppose that u find a nearby node D_i which is a leader of cluster C_i , and C_i has n_i members. u sends a message to join the cluster of D_i , and marks itself as a member of cluster C_i . Member addition algorithm [17] is process to get new group key, but this algorithm change the leader of the cluster. For the cluster leader saving same information that used to get group key, it is not a good choice to change the leader of the cluster. We modified that algorithm as follows:

1. We assume that the leader D_i saves the contents of messages he receives.
2. D_i sends $\{\alpha^{\prod\{r_k | k \in [1, n_i - 1] \wedge k \neq j\}} | j \in [1, n_i - 1]\}$, $\alpha^{r_1 \dots r_{n_i - 1}}$ to u .
3. u chooses a random exponent r_u and computes $\{\alpha^{r_u \prod\{r_k | k \in [1, n_i - 1] \wedge k \neq j\}} | j \in [1, n_i - 1]\}$, $\alpha^{r_u r_1 \dots r_{n_i - 1}}$, and sends them to the leader D_i .
4. D_i chooses an exponent r_{new} and computes $\{\alpha^{r_{new} r_u \prod\{r_k | k \in [1, n_i - 1] \wedge k \neq j\}} | j \in [1, n_i - 1]\}$, $\alpha^{r_{new} r_1 \dots r_{n_i - 1}}$, and broadcast to all the members in the cluster.

All the members in the cluster get the cluster key $\alpha^{r_{new} r_u r_1 \dots r_{n_i - 1}}$.

4.2 Key Refreshing

Key refreshing in mobile ad hoc networks is of great importance, because most nodes can be easily compromised due to their mobility and physical vulnerability. After the group key (the cluster key) is used for a period of time, the members in the group (the cluster) need refresh the group (the cluster) key in order to limit exposure due to the loss of group (cluster) session keys and limiting the amount of ciphertext available to cryptanalysis for a given group (cluster) session key.

Key refreshing protocol is based on the refreshment of member's key piece. Let D_i be the dominator of the cluster C_i and the random number he holds is r_D . When the cluster key should be refreshed, the key refreshing protocol is done as follows:

1. D_i chooses a new random number r'_D .
2. D_i computes $\{\alpha^{r'_D \prod\{r_k|k \in [1, n_i] \wedge k \neq j\}} | j \in [1, n_i]\}$, and broadcast to all the members in the cluster.

So now the key piece hold by the member in the cluster is $\alpha^{r'_D \prod\{r_k|k \in [1, n_i]\}}$.

When the group key should be refreshed, all the cluster leader process the protocol similar to above such that each of them gets the new group key. Then each cluster leader distributes the group key to all the members in the cluster.

4.3 Member Leaving

Deleting a mobile node M_i from the mobile ad hoc networks is also easy in our hierarchical key agreement protocol. Here, we assume that the leaving of node M_i will not disconnect the network.

If M_i is a member in the cluster C_l whose leader is node D_l , then D_l just apply member deletion [17] to get new cluster key for the cluster C_l . That is, D_l chooses a random number r_{new} and computes a new set of $n_l - 2$ sub-keys: $\{\alpha^{\prod\{r_j|j \in [1, n_l] \wedge j \neq k\}} | k \neq i\}$. Then he broadcast them to all cluster number so that all cluster number get a new cluster key. Next, all the cluster leaders process the key refreshing protocol to get the new group key, and each cluster leader distributes the new group key to all the members in the cluster.

Assume M_i is the leader of the cluster C_l . The protocol is done as follows:

1. M_{i-1} will take place of the leader of the cluster C_l and join to the backbone network.
2. All the member in cluster C_l using member deletion protocol in [17] to get a new cluster key.
3. The last member of the backbone network (D_g , if M_i is not the last member, and D_{g-1} , if M_i is the last number) sends $\{\alpha^{\prod\{R_k|k \in [1, g] \wedge k \neq j\}} | j \neq i\}$ to M_{i-1} , where R_k is the random number chosen by D_k at the key establishment phase.
4. M_{i-1} chooses a random number R' and computes $\{\alpha^{R' \prod\{R_k|k \in [1, g] \wedge k \neq j\}} | j \neq i\}$. Then he broadcast this to all the member in the backbone network so that all the cluster leaders get a new group key.
5. Each cluster leader distributes the group key to all the members in the cluster.

4.4 Analysis

When we execute the member joining, key refreshing or member leaving protocol, the first step is to find the route of the clusters and the backbone network. It can be done by lots of routing protocol such as [10] and the communication cost of the this step is $O(n)$.

After the key establishment, the nodes in the mobile ad hoc network moves around in the network. After a period of time, all the members in a cluster may disperse into the network, and we can view the cluster as the backbone networks. For the GDH protocol is used for the key agreement among each cluster nodes, we first find members in the cluster and then order the cluster nodes and connectors as $M_{i,1}, M_{i,2}, \dots, M_{i,n_i}$, such that the total communication cost from $M_{i,j}$ to $M_{i,j+1}$ ($1 \leq j < n_i$), is linear. Notice that there are n_i nodes totally. For simplicity, let $c(M_{i,j}, M_{i,j+1})$ be the communication cost from $M_{i,j}$ to $M_{i,j+1}$ in the network, i.e., the number of hops connecting them. We analyze the total communication cost $\sum c(M_{i,j}, M_{i,j+1})$ for the ordering derived by the above method. For the communication cost from $M_{i,j}$ to $M_{i,j+1}$ is linear, the total communication cost for broadcasting a message in a cluster is at most $\sum c(M_{i,j}, M_{i,j+1})$, which will be $O(n_i)$. Thus, the total communication cost for all cluster leader distributes the group key to the cluster members in the network is \sum_i (the total communication cost for broadcasting a message in a cluster C_i) $= \sum_i O(n_i) = O(n)$. Then, we can obtain the following table:

		joining	refreshing (group)	leaving (dominator)
comm.	hierarchical	$2c_1 + c_2$	$O(n) + c_2$	$c_1 + 2c_2 + O(n)$
cost	original GDH	$c_1 + n$	n	n
lasting	hierarchical	$2t_1 + t_2$	$2t_2$	$t_1 + 3t_2$
time	original GDH	$t_1 + t_2$	t_2	t_2
total	hierarchical	$3b$	$2g$	$2b + 2g$
exp.	original GDH	$2n$	$2n$	$2n$

From this table, we can see that our protocol does not increase much communication cost and lasting time compare with the original GDH protocol, and total exponentiation is much lower than the original GDH protocol.

5 Conclusion

In this paper, we propose a hierarchical key agreement protocol that is communication efficient to mobile ad hoc networks. We also show how to manage the group efficiently in mobile environments. Our protocol has following advantages:

1. The member can send encrypted messages to all using the group key and send encrypted messages to the members in its cluster using the cluster key.
2. The protocol is efficient during new members joining in, refreshing the group key and deleting the leaving members.

3. Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminals may vary with time. Our key agreement protocol does not maintain the topology of the network, and need not change the group key when the member are moving.
4. Because there are not many nodes in each cluster, it is easy to transmit a password and use the authenticated key agreement protocol.

Acknowledgements. This work is supported by the National Grand Fundamental Research 973 Program of China under Grant No. G1999035802; the Youth Foundation of the National Natural Science of China under Grant No. 60025205; the research collaboration program between Institute for InfoComm Research, Singapore and Institute of Software, Chinese Academy of Sciences. The authors would like to thanks the anonymous referees for their helpful comments.

References

1. N. Asokan and P. Ginzboorg, “Key-Agreement in Ad-hoc Networks”, *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, 2000.
2. G. Ateniese, M. Steiner and G. Tsudik, “Authenticated group key agreement and friends”, *ACM Conference on Computer and Communications Security — CCS’98*, pp. 17–26, 1998.
3. K. Alzoubi, P.-J. Wan and O. Frieder, “Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks”, *ACM International Symposium on Mobile Ad Hoc Networking and Computing — MobiHoc’02*, 2002.
4. M. Burmester, “On the risk of opening distributed keys”, *Advances in Cryptology — CRYPTO’94, LNCS 839*, Springer-Verlag, pp. 308–317, 1994.
5. Y. Desmedt and M. Burmester, “Towards practical proven secure authenticated key distribution”, *ACM Conference on Computer and Communications Security — CCS’93*, pp. 228–231, 1993.
6. M. Burmester and Y. Desmedt, “A secure and efficient conference key distribution system”, *Advances in Cryptology — EUROCRYPT’94, LNCS 950*, pp. 275–286, 1994.
7. K. Becker and U. Wille, “Communication complexity of group key distribution”, *ACM Conference on Computer and Communications Security — CCS’98*, pp. 1–6, 1998.
8. W. Diffie and M. E. Hellman, “New directions in cryptography”, *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.
9. M. Gerla and J. T. Tsai, “Multicenter, mobile, multimedia radio network”, *ACM-Baltzer Journal of Wireless Networks*, vol.1, no.3, pp. 255–265, 1995.
10. X. Hong, K. Xu and M. Gerla, “Scalable routing protocols for mobile ad hoc networks”, *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.
11. Y. Kim, A. Perrig and G. Tsudik, “Tree-based group key agreement”, *Cryptology ePrint Archive, Report 2002/009*, 2002.
12. Y. Kim, A. Perrig and G. Tsudik, “Simple and Fault-Tolerant Key Agreement For Dynamic Collaborative Groups”, 7th ACM Conference on Computer and Communications Security, pp. 235–244, 2000.

13. C. R. Lin and M. Gerla, "Adaptive clustering for mobile networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.
14. X. Li, Y. Wang and O. Frieder, "Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Networks", *IEEE International Conference on Computer Communications and Networks — ICCCN 2002*, 2002.
15. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
16. G. Pei, M. Gerla, X. Hong and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility", *IEEE Wireless Communications and Networking Conference — WCNC'99*, 1999.
17. M. Steiner, G. Tsudik and M. Waidner, "Diffie-Hellman key distribution extended to group communication", *ACM Conference on Computer and Communications Security — CCS'96*, pp. 31–37, 1996.
18. M. Steiner, G. Tsudik and M. Waidner, "CLIQUEs: A new approach to group key agreement", *International Conference on Distributed Computing Systems — 1998*, pp. 380–387, 1998.
19. P.-J. Wan, K. M. Alzoubi and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks", *Annual Joint Conference of the IEEE Computer and Communications Societies — INFOCOM 2002*, 2002.
20. Y. Wang and X. Li, "Geometric spanners for wireless ad hoc networks", *IEEE International Conference on Distributed Computing Systems — ICDCS 2002*, 2002.
21. K. Xu, X. Hong and M. Gerla, "An ad hoc network with mobile backbones", *IEEE International Conference on Communications — ICC 2002*, 2002.