A New Group Key Agreement Protocol for Ad Hoc Networks

Zhang Li-Ping
Lab. of information security,
College of Computer Science and Technology,
China University of Geosciences,
Wuhan, China
Carolyn321@163.com

Wang Yi
Department of information engineering,
Wuhan Police Vocation College,
Wuhan, China

***Abstract*—A mobile ad hoc network is a collection of autonomous nodes that communicate with each other by forming a multi-top wireless network. Different from conventional wireless networks, the resource of the nodes in ad hoc networks is limited and there may be tens of thousands of low-power energy constrained nodes in ad hoc networks. As such, the costs of the nodes resource and the network size should be taken into consideration when constructing a group key agreement protocol in the ad hoc networks. In this paper, an efficient and scalable group key agreement protocol based on layer-cluster group model for mobile ad hoc networks was proposed. In this protocol, multi-linear map is employed on layer-cluster structure to establish and allocate group key. So that it can not only meet security demands of larger mobile ad hoc networks but also improve executing performance.**

***Index Terms*—ad hoc networks, layer-cluster, group key agreement, multi-linear map**

I. INTRODUCTION

Wireless ad hoc networks are becoming progressively popular as they have the ability to form “on the fly” and can dynamically handle the joining or leaving of nodes in the network. However, the use of wireless links gives chances to attacks ranging from passive interception, replaying, and data interpolation, denial of service and identity forgery. In addition, wireless ad hoc networks usually operate in a wide open space and their topologies change frequently, so that the nodes are prone to be compromised. Because of these attacks, security measures should be adopted to protect the ad hoc communications.

Most security requirements, such as privacy authenticity and integrity, can be addressed by building upon a solid key management framework [1]. A secure group key agreement is the prerequisite for the security of these primitives, and thus essential to achieving secure infrastructure in ad hoc networks. However, the larger size of the group and the dynamic character of group changes pose a challenge on group key management research for wireless ad hoc networks.

Nodes in wireless ad hoc networks are usually low power devices that run on batter power and become unusable after failure or energy depletion. As a result, there is a need to employ energy-efficient group key agreement protocol in order to increase the overall network longevity.

Furthermore, given the potentially large number of mobile devices, scalability becomes another critical issue. The scalability problem can be solved by partitioning the communicating devices into subgroups, with a leader in each subgroup, and further organizing the subgroups into hierarchies [2].

In this paper we propose a new group key agreement protocol $LCML$, aimed at addressing a lightweight and fast solution in ad hoc networks. In protocol $LCML$, the network is partitioned into several clusters to construct h layers. On this layer-cluster model, multi-linear map is employed to establish group key which can not only meet security demands of mobile ad hoc networks but also reduce the communication costs.

The rest of this paper is organized as follows. In section II, we discuss related works on group key agreement protocols for ad hoc networks. Section III presents our key agreement protocol. In section IV, the security of the proposed protocol is discussed. We discuss the performance in section V, and conclude the paper in section VI.

II. RELATED WORKS

Recently, the area of group key management over

Manuscript received January 31, 2010; revised April 18, 2010; accepted May 11, 2010.

The Project was Supported by the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences （Wuhan） No. CUGL090246

© 2010 ACADEMY PUBLISHER
doi:10.4304/jcm.5.7.545-550

wireless ad hoc networks has received a significant amount of attention in literature. Since the foundational Diffie-Hellman (DH) protocol [3], several other protocols have been proposed for the group case. The first group key agreement protocol known as ING protocol was proposed by Ingemarsson et al. [4] in 1982. Following their work, Steiner et al. [5, 6, 7] proposed a family of protocols known as Group Diffie-Hellman (GDH.1, GDH.2 and GDH.3). In these protocols, the last group member servers as a controller and performs most of the computation on behalf of other group members in the group, therefore it needs more energy compared with other group members. Due to the limitation of the nodes energy the GDH protocol is inappropriate to the ad hoc networks. Kim et al. extended the work of a tree-based key agreement scheme by Perrig [8] to design a Tree-Based Group Diffe-Hellman (TGDH) protocol in [9]. Compared with GDH, it scales down the number of exponentiations and received messages required by the last group member to avoid excessive computational and communication costs required by one node. But TGDH protocol still requires each group member to perform large modular exponentiations and transmit/receive large messages. So the TGDH protocol is also inadequate for ad hoc networks. Kim et al. also proposed another tree-based key agreement scheme named as STR [10], which is quite similar to TGDH. In 2005, an efficient GKA protocol for low-power mobile devices was proposed by Cho et al [11]. However, this protocol requires a special member U_n to perform high computation on behalf of other members in the group. In the same year, Teo et al. [12] proposed an energy-efficient and scalable group key agreement scheme named as C-H protocol, which claimed that it is adapted to the large ad hoc networks. Although the C-H protocol logarithmically scales down the number of exponentiations, it increases the communication costs, compared to the GDH protocol and TGDH protocol. Based on their work, Zhang et al. [13] proposed a new protocol CH-ECC. In this protocol, the elliptic curve cryptosystem is employed by circular hierarchical group model to establish group key. So that it scales down the costs of communication. However the scalability problem is not taken into account in this protocol.

In order to solve the scalability problem, Jason H. et al. proposed a scalable key management and clustering scheme for ad hoc networks [2]. In this protocol, the communicating devices are divided into subgroups, with a leader in each group, and then organizing the subgroups into hierarchies. On this hierarchic structure, Diffie-Hellman protocol is used to establish group key. While this is one of the most recognized energy-efficient clustering protocols, its performance can be further enhanced.

Dan Boneh and Alice Silverberg studied some questions in linear algebra and cryptography and then presented several applications of multi-linear forms to cryptography [14].

Now, we give a definition of a d multi-linear map. Let G_1 be a cyclic additive group of prime order p and G_2 be a cyclic multiplicative group of same order p. We assume that the discrete logarithm problems (DLP) in both G_1 and G_2 are intractable. A map $e: G_1{}^d \rightarrow G_2$ is a d multi-linear map if it satisfies the following properties [14]:

1. Multi-linear: For $\forall a_1,...,a_d \in Z_p^*$ and $\forall P_1,...,P_d \in G_1^*$, $e(a_1 P_{1,...,} a_d P_d) = e(P_{1,...,} P_d)^{a_1 ... a_d}$

2. Non-degenerate: if $P \in G_1$ is a generator of G_1 then $e(P_{,...,} P)$ is a generator of G_2;

3. Computable: There exists an efficient algorithm to compute $e(P_{1,...,} P_d)$ for $P_1,...,P_d \in G_1^*$.

Based on their work [14], some group key management protocols were proposed [15, 16]. A common advantage of those protocols is that the one-round multi-party key exchange can be easily performed. In addition the security of those protocols always based on the Decisional Multi-linear Diffie-Hellman problem and Decisional Multi-linear Diffie-Hellman Assumption.

Definition1. The Decisional Multi-linear Diffie-Hellman (DMDH) problem is given $(P, a_1P, a_2P, ..., a_{d+1}P)$ and $z \in G_2$, to decide whether $z = e(P, P, ..., P)^{a_1 a_2 ... a_{d+1}}$ or not.

Definition2. Decisional Multi-linear Diffie-Hellman Assumption claims that for any polynomial time algorithm T and any $d > 1$, the advantage $DMDH_{T,d}(t)$ of T in solving the Decisional Multi-linear Diffie-Hellman problem is negligible, where $DMDH_{T,d}(t)$ is the probability that T can distinguish $e(P, P, ..., P)^{a_1 a_2 ... a_{d+1}}$ from $z \in G_2$.

Although Dan Boneh and Alice Silverberg point out those multi-linear maps is hard to build we believe that this issue can be solved by new techniques soon.

III. LAYER-CLUSTER KEY AGREEMENT PROTOCOL

A. Notation and Terminology

We use the following notation throughout the rest of this paper:

h: total number of layers in the group model;

L_i; *i*th layer for $i \in [0,\ldots,h\text{-}1]$ in the group model;

n: group size i.e. the total number of the nodes in the group model;

n_1: total number of subgroups when the group size is *n*;

tsg_{L_i} : total number of subgroups at layer L_i;

$SG_j^{(L_i)}$: *j*th subgroup at layer L_i ($j \in [0,\ldots, tsg_{L_i}\text{-}1]$);

$U_{SG_j}^{(L_i)}$: subgroup controller of the *j*th subgroup at layer L_i;

ub_{L_i} : the upper bound of the size of subgroup at layer L_i;

lb_{L_i} : the lower bound of the size of subgroup at layer L_i;

$t_{SG_j^{(L_i)}}$: total number of subgroup members in jth subgroup at layer L_i, $lb_{L_i} \le t_{SG_j^{(L_i)}} \le ub_{L_i}$;

$U_{(j,k)}^{(L_i)}$: kth member of L_i and it in subgroup $SG_j^{(L_i)}$ ($k \in [0,\ldots, \sum_{j=0}^{tsg_{L_i}-1} t_{SG_j^{(L_i)}}$ -1]);

$\{m\}_e$: a symmetric key encryption scheme;

B. Description of layer-cluster group model

In order to secure group communication for a large ad hoc network containing n users, the proposed protocol ($LCML$) adopt a layer-cluster group model as shown in Fig.1.

Denote the highest layer as L_0 while the lowest layer as L_{h-1}. In the layer-cluster group model each layer L_i ($i \in [0,\ldots,h$-1]) consists of subgroups denoted as $SG_j^{(L_i)}$ ($j \in$ [0,…, tsg_{L_i} -1]) and each subgroup $SG_j^{(L_i)}$ have some subgroup members denoted as $U_{(j,k)}^{(L_i)}$, in which k represents the position of the subgroup member at the layer L_i. The size of subgroup $SG_j^{(L_i)}$ is restricted by a lower and an upper bound. Each layer has one lower and upper bound which will be used across all the subgroups in that layer. And each layer can has different a pair of bound. Denote the minimum lb_{L_i} among all lb_{L_i} ($i \in$ [0,…,h-1]) as $lb_{\min}$ and the maximal ub_{L_i} among all ub_{L_i} ($i \in$ [0,…,h-1]) as $ub_{\max}$ in layer-cluster group model. Further the subgroups in each layer should be disjoint.

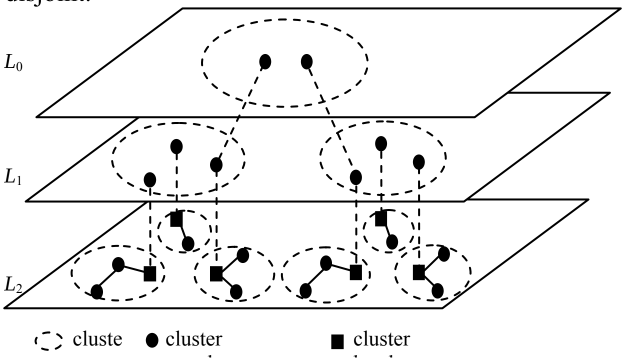

Fig.1. An illustration of the layer-cluster group model with h=3

In the layer-cluster group model a cluster is represented by a subgroup and a cluster member is represented by a subgroup member. In each subgroup all the subgroup members are arranged in a ring and let the subgroup member which represents the cluster head be the first member. Each subgroup $SG_j^{(L_i)}$ is managed by a subgroup controller $U_{SG_j}^{(L_i)}$ who is also the first member of that subgroup, i.e. $U^{(L_i)}_{(j,\sum_{s=0}^{j-1} t_{SG_s^{(L_i)}}+1)} = U_{SG_j}^{(L_i)}$ ($j \in$ [1,…, tsg_{L_i} -1]). The subgroup controller of all the subgroups in layer L_i except the highest layer $L_i \neq L_0$ join the layer L_{i-1}. So the subgroup members $U_{(j,k)}^{(L_i)}$ in each layer except the lowest layer $L_i \neq L_{h-1}$ are also subgroup controllers $U_{SG_k}^{(L_{i+1})}$ of subgroup $SG_k^{(L_{i+1})}$ at the next layer L_{i+1}, i.e. $U_{(j,k)}^{(L_i)} = U_{SG_k}^{(L_{i+1})}$.

C. Group key agreement protocol based on layer-cluster group model

In this section, we propose a new group key agreement protocol ($LCML$) based on layer-cluster group model for ad hoc networks. This protocol comprises three phases as follows:

$Phase1$: the proposed protocol $LCML$ starts at the lowest layer L_{h-1}. The process of subgroup key agreement in subgroup $SG_0^{(L_{h-1})}$ at the lowest layer L_{h-1} is described in details as follows:

1. Every subgroup member $U_{(0,k)}^{(L_{h-1})}$ of subgroup $SG_0^{(L_{h-1})}$ chooses an integer $r_{(0,k)} \in Z_p^*$ randomly as its private key.

2. Every subgroup member $U_{(0,k)}^{(L_{h-1})}$ computes its public key $r_{(0,k)}P$ and broadcast it to subgroup $SG_0^{(L_{h-1})}$.

3. After subgroup member $U_{(0,k)}^{(L_{h-1})}$ obtain all public keys of other subgroup members in $SG_0^{(L_{h-1})}$ it can compute subgroup key $K_{SG_0^{(L_{h-1})}}$ as follows:

$$K_{SG_0^{(L_{h-1})}} = e(r_{(0,0)}P,\ldots,r_{(0,k-1)}P, r_{(0,k+1)}P,\ldots,r_{(0,t_{SG_0^{(L_{h-1})}}-1)}P)^{r_{(0,k)}}$$

since

$$\begin{aligned} K_{SG_0^{(L_{h-1})}} &= e(r_{(0,1)}P, r_{(0,2)}P\ldots,r_{(0,t_{SG_0^{(L_{h-1})}}-1)}P)^{r_{(0,0)}} \\ &= e(r_{(0,0)}P, r_{(0,2)}P\ldots,r_{(0,t_{SG_0^{(L_{h-1})}}-1)}P)^{r_{(0,1)}} \\ &= e(r_{(0,0)}P,\ldots,r_{(0,k-1)}P, r_{(0,k+1)}P,\ldots,r_{(0,t_{SG_0^{(L_{h-1})}}-1)}P)^{r_{(0,k)}} \\ &= \ldots = e(P,P,\ldots,P)^{r_{(0,0)}r_{(0,1)}\cdots r_{(0,t_{SG_0^{(L_{h-1})}}-1)}} \end{aligned}$$

According to similar methods mentioned above, other subgroups at the lowest layer L_{h-1} can obtain their subgroup keys respectively.

$Phase2$: Each subgroup member $U_{(j,k)}^{(L_m)}$ of subgroup $SG_j^{(L_m)}$ at the layer L_m ($m \in [h$-2,…,0]) will run the subgroup key agreement protocol similar to Phase1 to obtain its subgroup key $K_{SG_j^{(L_m)}}$. Because the subgroup member $U_{(j,k)}^{(L_m)}$ is also the subgroup controller $U_{SG_k}^{(L_{m+1})}$ of subgroup $SG_k^{(L_{m+1})}$ at the next lower layer L_{m+1}, so each subgroup member $U_{(j,k)}^{(L_m)}$ possesses the subgroup key $K_{SG_k^{(L_{m+1})}}$ of the subgroup $SG_k^{(L_{m+1})}$ at the layer L_{m+1}. Therefore in the phase2 each subgroup member $U_{(j,k)}^{(L_m)}$ will use the hash value of the subgroup key $K_{SG_k^{(L_{m+1})}}$ as its private key to compute the subgroup key $K_{SG_j^{(L_m)}}$, instead of choosing a new integer randomly. This phase continues until all subgroup members $U_{(0,k)}^{(L_0)}$ in the subgroup $SG_0^{(L_0)}$ obtain the final group key $K = K_{SG_0^{(L_0)}}$.

$Phase3$: Each subgroup member $U_{(0,k)}^{(L_0)}$ at the highest layer L_0 encrypts the final group key $K = K_{SG_0^{(L_0)}}$ using the subgroup key $K_{SG_k^{(L_1)}}$ of subgroup $SG_k^{(L_1)}$ and broadcast $\{K\}_{K_{SG_k^{(L_1)}}}$ to its respective subgroup $SG_k^{(L_1)}$ at the layer L_1.

At the layer L_m for $m \in [1,\ldots, h\text{-}2]$ each subgroup $U_{(j,k)}^{(L_m)}$ first decrypts the encrypted message received from its subgroup controller $U_{SG_j}^{(L_m)}$ who belongs to a subgroup in the layer $L_{m\text{-}1}$ and concatenated its subgroup key $K_{SG_j^{(L_m)}}$ follow the decrypted message. Because the subgroup member $U_{(j,k)}^{(L_m)}$ is also the subgroup controller $U_{SG_k}^{(L_{m+1})}$ of subgroup $SG_k^{(L_{m+1})}$ at the next lower layer L_{m+1}, so each member $U_{(j,k)}^{(L_m)}$ can encrypts the message using the subgroup key $K_{SG_k^{(L_{m+1})}}$ and broadcast the encrypted message to its respective subgroup $SG_k^{(L_{m+1})}$ at the layer L_{m+1}. This process will end when all subgroup members $U_{(j,k)}^{(L_{h-1})}$ at the lowest layer $L_{h\text{-}1}$ have obtained the final group key and the corresponding subgroup keys by decrypting the encrypted message received from its subgroup controller $U_{(j,k)}^{(L_{h-1})}$ who belongs to a subgroup in the layer $L_{h\text{-}2}$.

D. Re-Keying Operations

1. Member joins. When a new node U_{n+1} wants to join the group and there existing a subgroup $SG_j^{(L_{h-1})}$ contains less than $ub_{L_{h-1}}$ subgroup members at the lowest layer $L_{h\text{-}1}$, then the node U_{n+1} join this subgroup $SG_j^{(L_{h-1})}$. This subgroup will run the subgroup key agreement protocol to get the new subgroup key. And the corresponding subgroups at the layer L_m ($m \in [h\text{-}2,\ldots,0]$) above this subgroup will also run the subgroup key agreement protocol to update their corresponding subgroup key. After updating of group key K, all the new subgroup keys and the new group key K will be broadcasted down the layers to corresponding subgroup members securely using symmetric key cryptography.

If all the subgroup $SG_j^{(L_{h-1})}$ at the lowest layer $L_{h\text{-}1}$ contains $ub_{L_{h-1}}$ subgroup members, then construct the layer-cluster group model again and run the $LCML$ protocol to establish and allocate new group key.

2. Member leaves. Let $U_{(j,k)}^{(L_{h-1})}$ be a subgroup member who wants to leave the subgroup $SG_j^{(L_{h-1})}$. In this subgroup, other subgroup members $U_{(j,t)}^{(L_{h-1})}$ ($t \neq k$), after receiving the leaving requirement from subgroup member $U_{(j,k)}^{(L_{h-1})}$, will delete the information of subgroup member $U_{(j,k)}^{(L_{h-1})}$ and run the subgroup key agreement protocol again to refresh the subgroup key $K_{SG_j^{(L_{h-1})}}$. Moreover, all the corresponding subgroup keys above this subgroup will be updated. And then all new keys will be broadcasted down to corresponding subgroup member securely.

IV. SECURITY ANALYSIS

The security of protocol $LCML$ is based on decisional multi-linear Diffie-Hellman assumption and the security of the symmetric key encryption scheme.

In the subgroup key agreement protocol, every subgroup member broadcast its public key to the subgroup. So every subgroup member can obtain other subgroup member's public keys in the subgroup to compute the subgroup key by using DMDH assumption. Obviously the security of subgroup key agreement is based on DMDH assumption. Assume that the adversary want to get the subgroup key, he need to extract the subgroup member's private key from its public key in which is equivalent to solving an instance of discrete logarithm problem. Obviously the adversary can not obtain any private key of subgroup member then it can not obtain the subgroup key.

In the group key agreement process the subgroup member $U_{(j,k)}^{(L_m)}$ of subgroup $SG_j^{(L_m)}$ at L_m ($m \in [h\text{-}2,\ldots,0]$) uses the hash value of the subgroup key $K_{SG_k^{(L_{m+1})}}$ as its private key to run the subgroup key agreement protocol. This process will end when the subgroup at the highest layer L_0 has computed its subgroup key K . Obviously, based on the security of subgroup key agreement protocol an adversary will not be able to obtain the subgroup key $K_{SG_k^{(L_{m+1})}}$ and he will not be able to get the subgroup key $K_{SG_j^{(L_m)}}$ at L_m too.

In the protocol $LCML$, the final group K and the respective corresponding subgroup keys are encrypted and broadcasted down the layers to corresponding subgroup members using symmetric key cryptography. If the symmetric key encryption scheme is secure against chosen ciphertext attacks, then the adversary will not be able to obtain the group key unless he is able to successfully break the secure encryption scheme.

Theorem1. Protocol $LCML$ provides forward secure and backward secure.

Proof. *forward secure*: Let A be an active adversary who has been a member of some subgroup during some previous time period. Now assume the adversary A tries to read the subgroup traffic after he has left. A has with it the old group key and a series of corresponding subgroup keys. However, he can not read the subgroup traffic, since the protocol updates group key and all corresponding subgroup keys that A previously knows. So the adversary A can not read the subgroup traffic after he has left unless he join the subgroup again which provide the forward secure.

backward secure: In *LCML* protocol, when A joins a subgroup, this subgroup and all the corresponding subgroup above this subgroup will update their subgroup keys so the adversary A cannot derive any previous subgroup key and previous group key before he join the subgroup. Then the adversary A can not read the previous subgroup traffic before he joins the subgroup since he does not know any previous subgroup keys and the group key. According to the analysis above, the protocol *LCML* provides the backward secure.

V. COMPLEXITY ANALYSIS

We compared the computational overhead and communication costs of our proposed protocol with TGDH, GDH and C-H [12]group key agreement protocols. In Table 1, the computational overhead refers to the number of modular exponentiations and the number of DMDH operations required to compute the final group key and the communication cost is represented by the number of messages transmitted and received. Furthermore, as mentioned in [17, 18], compared with the computational overhead of symmetric key cryptography, the computational overhead of modular exponentiations are several orders of magnitude higher. So we neglect the computational complexity of symmetric key encryption/decryption as compared to modular exponentiat -ions. In table 1, the notation c refers to the number of members in each subgroup in protocol C-H and the notation h is presented the number of layers or the height of the tree. Furthermore, the users refer to the all subgroup members in each subgroup across the lowest layer L_{h-1}.

For TGDH protocol, it requires each user to perform $2h$ modular exponentiations, send and receive h messages respectively. In the TGDH protocol the height of the tree is $h = \log_2 n$ however the number of layers in the *LCML* protocol is $h \le \log_{lb_{\min}} n_1$. For example, for a group size n=2^{20}=1048576, the height of the TGDH tree is $h = \log_2 2^{20} = 20$ while the number of layers in *LCML* protocol is $h \le \log_8 2^{15} = 5$ with n_1= 32768, $lb_{\min}$=8 and $ub_{\max} > 32$. Compared with TGDH protocol, the proposed protocol *LCML* reduces the computation cost and the messages need to send.

As shown in Table 1, the C-H protocol requires each user to perform three modular exponentiations, transmit two messages and receives c+2 messages. A subgroup member in the $SG_j^{(L_m)}$ ($m \in$[1,…,h-2]) has to compute 3(h-m) modular exponentiations, transmit 3h-3m-1 messages and receive (h-m)(c+1)+1 messages respectively. While a subgroup member in the $SG_0^{(L_0)}$ requires to perform 3h modular exponentiations, send 3h-1 messages and receive h(c+1) messages. Compared with TGDH, the C-H protocol scales down the number of exponentiations and the transmitted messages, but increases the number of received messages. In our proposed protocol, each user requires to perform one modular exponentiation and one DMDH operation, transmit one message and receive less than $ub_{L_{h-1}}$ messages. A subgroup member in the $SG_j^{(L_m)}$ ($m \in$[1,…,h-2]) has to compute h-m modular exponentiations and h-m DMDH operations, transmit 2(h-m)-1 messages and receive less than $ub_{\max}(h-m)$ messages respectively. While a subgroup member in the $SG_0^{(L_0)}$ requires to perform h modular exponentiations and h DMDH operations, send 2h-1 messages and receive less than $(ub_{\max})h-1$ messages. Obviously, our proposed protocol *LCML* scales down the communication costs as compared to the C-H protocol.

TABLE 1

COMPUTATION AND COMMUNICATION COST

		Exp.	DM DH	Messages sent	Messages to be received
TGDH		$2h$	0	h	h
GDH	U_1-U_{n-2}	3	0	2	3
	U_{n-1}	2	0	1	2
	U_n	n	0	1	n-1
C-H	$SG_0^{(L_0)}$	3h	0	3h-1	h(c+1)
	$SG_j^{(L_m)}$ $m \in$ [1,⋯,h-2]	3(h-m)	0	3(h-m)-1	(h-m)(c+1)+1
	users	3	0	2	c+2
LCML	$SG_0^{(L_0)}$	h	h	2h-1	$\le (ub_{\max})h-1$
	$SG_j^{(L_m)}$ $m \in$ [1,…,h-2]	h-m	h-m	2(h-m)-1	$\le ub_{\max}(h-m)$
	users	1	1	1	$\le ub_{L_{h-1}}$

VI. CONCLUSION

In ad hoc networks, secure group key agreement protocols play a key role. They are one of the most crucial technologies for ad hoc networks. However, most existing group key agreement protocols require either centralized key servers or expensive public key operations, which make them unsuitable for ad hoc networks. In this paper, we proposed a new group key agreement protocol based on DMDH assumption and layer-cluster group model. Compared with TGDH, GDH and C-H group key agreement protocols, the proposed protocol *LCML* improve the executing performance. So it is more suitable for ad hoc networks.

ACKNOWLEDGMENT

Thanks for the helpful discussion about the proof of security from PhD Candidate XuPeng and PhD MingHui Zheng.

REFERENCES

[1] G.Ateniese, M.Steiner and G.Tsudik. New multi-party authentication services and key agreement protocols. IEEE Journal on Selected Areas in Communications, 2000, 18(4):628-640

[2] Jason H. Li, Renato Levy, Miao Yu. A Scalable Key Management and Clustering Scheme for Ad Hoc Networks. In: INFOSCALE'06, 2006, 1-10

[3] W.Diffie, M.Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 1976, 22: 644-654

[4] I.Ingemarsson, D.T.Tang and C.K.Wong, A conference key distribution system. IEEE Transactions on Information Theory, 1982, 28(5): 714-720

[5] Steiner M, Tsudik G., Waidner M. Differ-Hellman key distribution extended to group communication. In: Usenix Conference on Computer and Communications Security, ACM Press, 1996, 31-37

[6] Steiner M, Tsudik G, Waidner M. DLIQUES: A New Approach to Group Key Agreement. In: Proceeding of the 18th International Conference on Distributed Computing Systems, 1998, 380-387

[7] Steiner M, Tsudik G, Waidner M. Key Agreement in Dynamic Peer Groups. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(8): 769-780

[8] A.Perrig. Efficient collaborative key management protocols for secure autonomous group communication. International Workshop on Cryptographic Techniques and Electronic Commerce, 1999, 192-202

[9] Kim Y, Perring A, Tsudik G. Tree-Based Group Key Agreement. ACM Transaction on Information and System Security, 2004, 7(1): 60-96

[10] Kim Y, Perring A, Tsudik G. Group Key Agreement Efficient in Communication. IEEE Transaction on Computers, 2004, 53(7): 905-921

[11] S. Cho, J. Nam, S.Kim and D. Won. An Efficient Dynamic Group Key Agreement for Low-Power Mobile Devices. ICCSA'2005. 2005,3480:498-507

[12] Joseph Chee Ming Teo, Chik How Tan. Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks. PE-WASUN'05, 2005, 114-121

[13] Zhang Li-Ping, Cui Guo-Hua, Yu Zhi-Gang. An Efficient Group Key Agreement Protocol for Ad Hoc Network. WICOM08, October, Dalian, China, 2008

[14] D.Boneh and A.Silverberg. Applications of multi-linear forms to cryptography, Contemporary .Mathematics, 2003, 324:71-90.

[15] Wei Wang, Jiangfeng Ma and SangJae Moon. Efficient Group Key Management for Dynamic Peer Networks. MSN2005, 2005, 3796:753-762.

[16] Zhang Li-Ping, Cui Guo-Hua et al. Group Key Agreement Protocol Based on Circular Hierarchical for Ad Hoc Network. Computer Science, 2008,35(10):61-64

[17] Carman D.W., Kruss P.S., Matt B.J. Constraints and approaches for distributed sensor network security. NAI Labs Technical Report, 2000.

[18] Trappe W, Wang Y, Liu K.J. Resource-aware conference key establishment for heterogeneous networks. IEEE/ACM Transactions on Networking, 2005, 13(1): 134-146.

Zhang Li-Ping, born in 1978, Ph.D. Her research interests include public key cryptography, provable security of cryptographic protocol and cryptography theory and practice, Ad hoc networks.
Email:carolyn321@163.com

Wang Yi, born in 1975, Master. Her research interests include information security, algorithm analysis and numeric analysis.