International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 46-51, 2018. ISSN (Online): 2456-7361

Scalable Group Key Agreement using Key Tree in Mobile Environment

Seokhyang Cho[1]
[1]Department of Information and Communication, Pyeongtaek University, Pyeongtaek, Republic of Korea, 17869

Abstract*—This paper proposes an efficient group key agreement protocol in a mobile environment where group members frequently join and leave. This protocol consists of basic protocols and general ones and is expected to be suitable for communications between a mobile device with limited computing capability and a key distributing center (or base station) with sufficient computing capability. Compared with other schemes, the performance of the proposed protocol is a bit more efficient in the aspects of the overall cost for both communication and computation where the computational efficiency of the scheme is achieved by using exclusive or operations and a one-way hash function. Also, in the aspect of security, it guarantees both forward and backward secrecy based on the computational Diffie-Hellman (CDH) assumption so that secure group communication can be made possible.*

***Keywords*—** *Computational Diffie-Hellman assumption: group key agreement: hierarchical key tree: multicast.*

I. INTRODUCTION

As universal electronic transaction using mobile devices has been quite prevalent in an internet environment, wireless communication technology has spread widely in everyday life. Based on the contents of mobile devices, the mobile computing technology supports the transaction systems by configuring dynamically the local and remote network so that they can share information in real time. The parties communicating to share information consist of a fixed server (called an application server or service provider) with sufficient computing power and mobile devices, called clients, with limited computing resources. In terms of computing power, such an asymmetric mobile environment is common in many applications such as collaborative work via networks, multi-user games, internet stock quotes, audio and music transfer, pay TV program service depending on watched program, and software updates [1-4].

With the development of group-oriented applications, the necessity of security services for secure group communication is also rapidly increasing. Especially, a mobile environment has many security weaknesses such as eavesdropping, data forgery, alteration, and destruction, and illegal use of data. Thus, if information is not properly protected, confidential information can be exposed to unauthorized users. Especially, if financial and economic information is leaked, there is also a risk of financial loss. Thus, secure group communication can be effectively accomplished by sharing a single secret key, called a group key, among all members of a group.

For example, suppose one member in a group sends a secret message to all the other members of the group. If all the members of the group share a group key, the sender uses the group key to encrypt the message once and sends it, all the recipients in the group can use the same group key to decrypt the received message safely. Therefore, a method to securely and efficiently share a group key among group members is required. A protocol designed for this purpose is called a group session key establishment protocol.

This common session key should be shared by group members in a secure manner over an open network. The protocol for establishing a session key sets a common secret key known only to members participating in the communication and exchanges public information so that no member can determine the key in advance. In addition, in the session key setting, forward secrecy must be provided so that a member having left the group could not know a new session key from a part of the session key previously shared after he/she has withdrawn and also backward secrecy must be guaranteed so that a new member having joined the group could not know the previous session key from a new session key.

The protocol for establishing a key can be categorized into key transport protocols and key agreement protocols from the view point of generating the session key. The key agreement protocol is a scheme in which one or more members present their own information for creating a common session key while the key transport protocol is a scheme in which a member creates a session key and transmits it to other members in a secure manner. In particular, a scheme is said to be a contributory key agreement protocol if all members participating in the protocol provide their information for the session key establishment. In contrast, if all members, having the same structure, transmit messages and perform calculations, it is said to be role symmetric. The scheme proposed in this paper is an asymmetric contributory key agreement protocol requiring a less computational complexity for mobile devices. It has a feature that as long as the randomness of the number selected by the group members is ensured, the session key value cannot be controlled by them even if they make a conspiracy.

Key agreement is one of the basic elements of cryptography. Diffie-Hellman [5] proposed the first protocol on key agreement, but it was under the man-in-the-middle attack because it has no way to authenticate communicating parties. After the scheme, many protocols have solved this problem by combining a digital signature with key agreement protocols.

In this paper, we propose an effective group key agreement protocol based on the Diffie-Hellman assumption. The proposed protocol is suitable for dynamic groups because it

International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 46-51, 2018. ISSN (Online): 2456-7361

has a high computational efficiency on the client side and low computation cost for group membership withdrawal or subscription. The proposed protocol provides perfect forward secrecy and an improved computational efficiency by using a hash function and exclusive or operations.

In Section 2 of this paper, we review existing related works, and in Section 3, we introduce a basic group key agreement protocol applied for the proposed key tree and a protocol generalized by applying the basic protocol for a subgroup of the key tree. In Section 4, we compare and analyse the performance of the existing protocol and the proposed protocol in terms of computational complexity and communication complexity. Finally, we make a conclusion in Chapter 5.

II. Related Work

A. Yonadae Kim et al.'s Protocol

Yongdae Kim et al. [6] proposed an efficient group key agreement protocol in terms of communication cost rather than computation cost. This protocol is suitable for high-latency networks, and even if the network fails, it can communicate except for the failed node. The reason for this is that the problem can be solved by deleting the failed node and then updating the key tree. It is also a feature of the group key agreement protocol using the key tree which makes possible the joining and leaving operations of group members.

This protocol based on the key tree computes the group key by hiding the secret key of each member participating in the communication as the exponentiation on the modular operation. Since the group members can compute the secret session key as the exponential power of base, the efficiency of the protocol in terms of computation cost is relatively low compared with other protocols using one-way hash function and exclusive or operation.

As shown in Fig. 1, the group key, $K_{<0,0>}$, corresponding to the root node of the key tree composed of 7 group members, M_i (i =1, 2, …,7), is determined by Eq. (1):

$$K_{<0,0>} = \alpha^{(\alpha^{\alpha^{r3} r4} \alpha^{r1 r2})(\alpha^{\alpha^{r5} \alpha^{r6} r7})} \quad (1)$$

where the node <l, v> represents the v-th node at the level l of the tree, r_i is the random secret key of M_i, and α is the base value of the exponentiation.

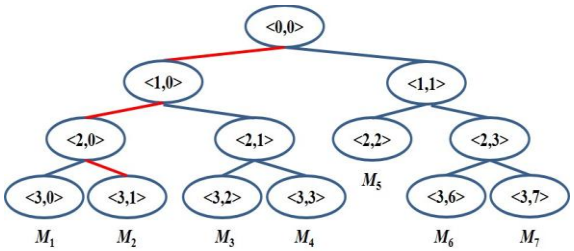

Fig. 1. Example of Yongdae Kim et al.'s key tree.

For example, in Fig. 1, the member M_2 can determine the key values $K_{<2,0>}$, $K_{<1,0>}$, $K_{<0,0>}$ existing on its path from the lowest node to the root node by using the internal node of the subtree to which it belongs, the blinded secret keys $BK_{<3,0>}$, $BK_{<2,1>}$ of the sibling node, the blinded key $BK_{<1,1>}$ of the next subtree to which it does not belong, and its own secret key $K_{<3,1>}$ where the value of the blinded key is computed as $\alpha^{r_i} \bmod p$ (p is a very large prime number).

B. Sangwon Lee et al.'s Protocol

Sangwon Lee et al. [7] modified the TGDH (Tree-based Group Diffie-Hellman) protocol into a pairing-based bilinear map and applied it for a ternary key tree to propose a group key agreement protocol that improved computational efficiency, keeping the performance of TGDH in terms of communication complexity.

As shown in Fig. 2, the group key $K_{<0,0>}$ corresponding to the root node of the key tree composed of 7 group members M_i (i =1, 2, …,7) is computed as follows:

$$K_{<0,0>} = H_1(\hat{e}(H_1(\hat{e}(P,P)^{r_4 r_5 r_6})P, r_7 P)^{H_1(\hat{e}(r_1 P, r_2 P)^{r_3})}) \quad (2)$$

where P is given as a point on an elliptic curve, H_1 is the hash function from codomain G_2 to Z_q^* in the bilinear map, $\hat{e}(P,P)$ is the generator of G_2, r_i is an element of Z_q^* of member M_i that represents a secret key value randomly selected.

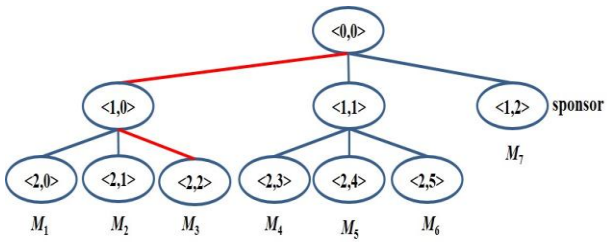

Fig. 2. Example of Sangwon Lee et al.'s key tree.

For example, in Fig. 2, the member M_3 can determine the key values $K_{<1,0>}$, $K_{<0,0>}$ existing on its path by using the blinded key $BK_{<2,0>}$, $BK_{<2,1>}$ of the sibling node, the blinded $BK_{<1,1>}$ of the subtree to which it does not belong, and its own secret key $K_{<2,2>}$ where the value of the blinded key is computed as $BK_{<l,v>} = K_{<l,v>} P$.

C. EHBT: Sandro Rafaeli et al.'s Protocol

Sandro Rafaeli et al. [8] proposed a group key management protocol that uses efficient hierarchical binary tree (EHBT) to reduce the size of the key update message, not increasing storage space and processing requirements compared with other HBT protocols.

In the EHBT protocol, the key distribution center (KDC) acts as a server maintaining the key tree and the group members, as clients, have the KEKs (Key Encryption Keys) associated with the leaf node and the KEKs of the ancestors on the path from the lowest node to the root node of the tree.

The keys are generated from other keys, by using the one-way hash function h and the exclusive or operation as

$$F(x, y) = h(x \oplus y) \quad (3)$$

where h hides the two values by passing the result through the one-way function and the exclusive or operation mixes the two values to generate a new value.

When the member M_i changes the key k_i into a new key k_i' or the member M_j needs to update the key k_i, they are provided

International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 46-51, 2018. ISSN (Online): 2456-7361

with the current key k_i and the index i serving as an identifier or the key of the left or right child node of i together with the index as the input information to the key generating function, as expressed by the following equation:

$k_i' = F(k_i, i)$ or $k_i' = F(k_i^{\text{left | right}}, i)$ (4)

For example, if a group member M_1 is assigned to a leaf node of a binary tree, and the M_2 wants to participate in this group communication, the KDC updates the tree as shown on the right in Figure 3. Then, the new member M_2 is located at the leaf node n_2 and allowed to get a randomly selected key k_2 from the KDC. Also, node n_{12} is inserted into the tree as the parent node of n_1 and n_2 where the keys belonging to the set of ancestor nodes of n_1 in the route from the lowest node to the root node are changed to new values as follows.

$$\begin{cases} k_1' = F(k_1, 1),\ K_{12} = F(k_1', 2) \\ K_{14}' = F(K_{14}, 14),\ K_{18}' = F(K_{18}, 18) \end{cases} \quad (5)$$

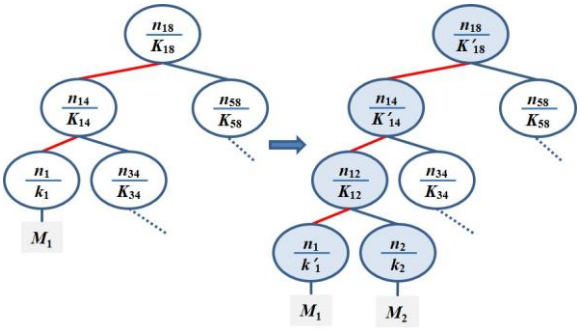

Fig. 3. Example of Sandro Rafaeli et al.'s key tree.

In addition to the protocols mentioned above, there are also studies on group key agreement protocols [9-11] applying the points on elliptic curves to a key tree and several studies [12-14] that create the secret value session key by exponentiating the generator element based on the computational Diffie-Hellman assumption.

In this paper, we propose a group key agreement protocol that, based on computational Diffie-Hellman assumption, uses a hash function and exclusive or operations to improve the efficiency not only in terms of computational complexity but also communication cost in comparison with other protocols.

III. The Proposed Protocols

Table I shows the system parameters used in this paper.

TABLE I. Notations.

Parameters	Details
d	Height of the tree
n	Number of members in the group
m	Number of leaving members in the original group
P	Large prime number
g	Generator in the group Z_p^*
G	Subgroup of prime order q in Z_p^*
M_i	i-th group member
PK_i	Signature verification key of M_i
$Sign$	Signing algorithm
H	Hash function
SK	Shared session key among members

A. Basic Protocol using Ternary Key Tree

In this section, we describe the basic protocol *BS* consisting of KDC corresponding to the root node and the group members for the case where the depth of the tree is 1.

Table I shows the system parameters used in this paper. Let L_0, L_1, and L_2 be a group of members at levels 0, 1, and 2, respectively, that is, $L_0 = \{S_1\}$, $L_1 = \{M_1, M_2, M_3\}$, and $L_2 = \varnothing$ where S_1 is the key distribution center playing as the server and M_i (i = 1, 2, 3) are group members corresponding to the leaf nodes. The members participating in group communication are shown in Fig. 4.

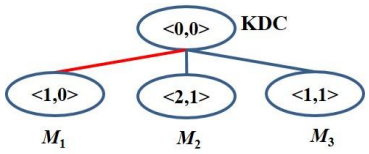

Fig. 4. Example of basic protocol's key tree.

It is assumed that the public parameter, being a large prime p, and the generator element g, being the basis of the exponentiation, are known to all group members participating in the communication. The execution procedure of the proposed protocol is as follows.

Step 1: Each member M_i selects a random number r_i in Z_q^*, exponentiates it with base g to determine the blinded value $z_i (= g^{r_i} \bmod p)$, signs it with its secret key K_i, and sends $\sigma_i = Sign_{K_i}(z_i)$ and the message $m_i = (z_i \| \sigma_i)$ to the server (KDC) where the server KDC also selects its secret key random number r_0 and s from Z_q^* and calculates z (= g^s mod p) and $x_0 = z^{r_0} (= g^{sr_0})$.

Step 2: The server having received the message m_i from each member M_i verifies the signature σ_i with the public key PK_i of each member, and then calculates the value of required for determining the session key of each member. Then, the shared common value X including the ID_i that can be used to identify the group members participating in the communication is calculated as

$$X = \bigoplus_{i=0}^{k} H(ID_i \| x_i) \quad (k\text{: the number of leaf node}) \ldots (5)$$

Also, if the set of X_i (= $X \oplus H(ID_i \| x_i)$) necessary for member M_i is $Y(=\{X_1, X_2, X_3\})$, the server signs it with the server's secret key K_s to make $\sigma_s (= Sign_{K_s}(I \| z \| Y))$ and multicasts the message $m_s (= (I \| z \| Y \| \sigma_s))$ to the group members where I is a set of ID_i.

Step 3 (Key Agreement Step): To calculate the common key, each group member verifies the signature σ_s and then restore the shared common values X and x_i as

$$X = X_i \oplus H(ID_i \| x_i),\ x_i = z^{r_i} \quad (6)$$

Now, all members of the group, including the server, can use a one-way hash function to calculate the common session key SK (= H ($X \| Y$)).

As shown in the execution procedure of the proposed basic protocol, the session key is shared with $n-1$ unicasts and one

International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 46-51, 2018. ISSN (Online): 2456-7361

multicast when the number of group member including the server is *n*.

B. Generalized Protocol

In this section, we describe a protocol generalized by applying the basic protocol to each subtree.

Let L_0, L_1, and L_2 be a group of members at levels 0, 1, and 2, respectively, that is, $L_0 = \{S_0\}$: KDC, $L_1 = \{S_1, S_2, S_3\}$, and $L_2 = \{M_1, M_2, \ldots, M_8\}$ where S_j (j = 1, 2, 3) is the base station serving as the server in each subtree and M_i (i = 1, 2, …, 8) is a group member corresponding to the leaf node. Each member participating in the group communication is shown in Fig. 5.

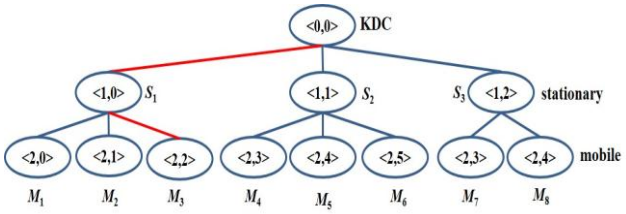

Fig. 5. Example of generalized protocol's key tree.

The execution procedure of the proposed protocol is as follows.

Step 1: Each member M_i in L_2 selects a random number r_i in Z_q^*, exponentiates it with base g to determine the blinded value $z_i (= g^{r_i} \bmod p)$, signs it with its own private key K_i, and sends $\sigma_i = Sign_{K_i}(z_i)$ and message $m_i = (z_i \| \sigma_i)$ to the internal node and the KDC, as is the same with the configuration step 1 of the basic protocol. Similarly to M_i, the base station S_j in L_1 also selects a random number r_{s_j} in Z_q^*, exponentiates it with base g to determine the blinded value, $z_{s_j} (= g^{r_{s_j}} \bmod p)$ signs it with its own secret key K_{s_j} to make $\sigma_{s_j} = Sign_{K_{s_j}}(z_{s_j})$, and sends $m_{s_j} = (z_{s_j} \| \sigma_{s_j})$ to the root node KDC where the base station selects s_j and the server KDC selects its own secret key random number r_0, s in Z_q^*, and they calculate $\{ z_{s_j} (= g^{s_j} \bmod p),\ x_{s_j} = z_{s_j}^{r_{s_j}} (= g^{s_j r_{s_j}}) \}$ and $\{ z (= g^s \bmod p),\ x_0 = z^{r_0} (= g^{sr_0}) \}$, respectively.

Step 2: Upon receipt of the message from member M_i corresponding to the child node from the level 2, each base station S_j verifies the signature σ_i with the public key PK_i of each member, and then calculates $x_{s_j} (= z_i^{s_j} \bmod p)$ required for determining the subgroup key corresponding to a subgroup of the key tree. Then each subgroup members compute the shared common value X_{s_j} including I_j that can be used to identify each member, as follows:

$$X_{s_j} = \bigoplus_{j=0}^{k} H(ID_j \| x_{s_j}) \quad (k\text{: the number of leaf nodes}) \tag{7}$$

where $x_{s_j} = (g^{rr_i})^{s_j}$, r is secret information of the KDC, and s_j is the secret information of each base station. Also, if the set of $X_{s_j} \oplus H(ID_j \| x_{s_j})$ necessary for the group members is Y', the server signs it with the secret key K_{s_j} of the base station to make $\sigma_{s_j} (= Sign_{K_{s_j}}(I_j \| z_{s_j} \| Y'))$, and multicasts the message $m_{s_j} (= (I_j \| z_{s_j} \| Y' \| \sigma_{s_j}))$ to the group members where I_j is the set of ID_j. The server KDC is the same with the configuration step 2 of the basic protocol having the base station as a member.

Step 3 (Key Agreement Step): To calculate the common key, each group member verifies the signature σ_{s_j} and then restore the shared common values X_{s_j}, x_{s_j} as

$$X_{s_j} = X_i \oplus H(I_i \| x_{s_j}),\ x_{s_j} = z_{s_j}^{r_i} \tag{8}$$

Now, all members of the group, including the server, can use a one-way hash function to calculate the common session key $SK (= H(X_{s_j} \| Y'))$.

As shown in the execution procedure of the proposed basic protocol, the common session key is shared with $n-1$ unicasts and $\lceil (n-1)/3 \rceil + 1$ multicasts when the number of group member including the server is *n*.

IV. Evaluation and Security Considerations

A. Evaluation of the Proposed Protocol

The proposed key agreement scheme needs $n-1$ (the number of group members excluding KDC) unicasts since the secret key of each member is secretly transmitted to the KDC (or base station) for setting the session key in the configuration phase 1. It also needs one multicast communication in the configuration phase 2. Therefore, the proposed protocol is executed with only two rounds of communication and it is optimal in terms of the number of necessary messages, which is *n* [15].

When a member leaves the group, the KDC (or base station) requires one multicast communication, and if a new member participates in the group communication, it needs as many unicasts as the number of the participating members and one multicast.

In the proposed key agreement scheme, each member M_i except KDC performs one signature generation and operations of two modular exponentiation (g^{r_i}, z^{r_i}). Then, the KDC (or base station) performs $n-1$ signature verifications and $n+1$ modulo operations of exponentiation to verify the signature of member M_i. This is because it is necessary to calculate $x_i (= (g^{r_i})^s)$ for all members excluding itself and the secret values g^{r_0} and $x_0 (= (g^{r_0})^s)$ of the KDC itself.

In addition, when a member leaves the existing group, TABLE original members need only a hash function calculation and exclusive or operations because they can use the previously calculated modular exponentiation value as it is. When a new member participates in a group communication, only new members need two modular operations of exponentiation and the KDC requires as many modular operations of exponentiation as new members.

International Journal of Scientific Engineering and Science
Volume 2, Issue 1, pp. 46-51, 2018. ISSN (Online): 2456-7361

B. Evaluatin of Other Protocols

Let n, d, and m be the number of original group members, the depth of the updated tree, and the number of members leaving the group, respectively.

In Yongdae Kim et al. [6], the number of exponentiation operations is $3n-7$ or $3n-3m+2$ depending on whether the number of members leaving the group is less than 3 or not, and therefore, the average operation cost is $3(n/2) + 2$.

In addition, in [6], when a new member joins the existing group, the communication cost is two rounds to be taken for all members to get the group key where in the first round, the new member broadcasts its own exponentiation operation value and in the second round, all the group members update the key tree and the sponsor creates its new random shared value, computes a blinded secret key value, and then broadcasts the updated key-value. On the other hand, when a member leaves a group, it takes just a single round for all members to get the group key since only the sponsor creates its new shared value, calculates all secret and blinded keys, and broadcasts the updated tree.

In [7] (Sangwon Lee et al.), when a member joins or leaves a group, the number of required pairing calculations is $\lceil \log_3 n \rceil - 1$ where $\lceil \log_3 n \rceil$ is the depth of the ternary tree whose leaf nodes all group members can be assigned to. Also, the number of multiplicative operations required for a point P on an elliptic curve, such as $2P(=P+P)$, $3P(=2P+P)$ and so on, is $\lceil \log_3 n \rceil + 1$. However, according to the work of Barreto et al. [16], this pairing operation takes about three times as long as an exponential multiplication on the modulus, which leads to lower efficiency in terms of computation cost than other protocols. And the communication cost in [7] is the same with that of protocol [6] because it updates the key tree and the secret value of the sponsor in the same way as in [6].

In the aspect of computational cost of EHBT [8], the total number of hash function computations is $3n+d$ since the KDC needs $3n-1$ times and the sibling node of the newly joining member needs $d+1$ times. Also, when half of the group members leave the group, the total number of hash function computations is $2n+d$ since the KDC needs $2n-1$ times and the sibling nodes of the members whose tree is updated and who leaves needs $d+1$ times. And the number of required exclusive or operations is the same as that of hash function computations since the input value of the hash function is calculated using a exclusive or operation.

Also, while the security of the EHBT protocol depends on the cryptographic property of the one-way hash function h, the proposed protocol is secure against passive attackers based on the CDH assumption as well as the one-way nature of the hash function.

The communication cost of EHBT for a joining member is a total of $2n-1$ multicasts since n unicasts are needed for each of the joining member and its sibling node and also $n-1$ mulicasts for the KDC. On the other hand, the communication cost for a leaving member is a total of $2n-1$ multicasts since n multicasts are needed for the sibling node of the leaving member and $n-1$ mulicasts for the KDC.

Table II summarizes the comparison between the proposed protocol and the three protocols discussed in the related researches in terms of communication cost and computation cost. It shows that the proposed protocol reduces the number of message transmissions by one and also reduce the number of modular exponentiations so that the efficiency has been improved in terms of both communication cost and computation costs through the use of hash function and exclusive or operations.

TABLE II. Communication and computation costs.

	Evaluation	Yongdae Kim et al. [6]		Sangwon Lee et al. [7]		EHBT [8]		Prosed protocol	
		Join	Leave	Join	Leave	Join	Leave	Join	Leave
Communication	Rounds	2	1	2	1	2	1	2	1
	Messages	3	1	3	1	$\log_2 n$	$\log_2 n$	2	1
Computation	Hash functions	0	0	$\lceil \log_3 n \rceil + 1$	$\lceil \log_3 n \rceil + 1$	$3n+d$	$2n+d$	$2(n-1)$	$2m$
	XOR operations	0	0	0	0	$3n+d$	$2n+d$	$2(n-1)$	$2m$
	Exponentiations(or [7] Point multiplications	dn	$\frac{3n}{2}+2$	$\lceil \log_3 n \rceil + 1$	$\lceil \log_3 n \rceil + 1$	0	0	$n+1$	m
	Pairings	0	0	$\lceil \log_3 n \rceil - 1$	$\lceil \log_3 n \rceil - 1$	0	0	0	0

C. Security Considerations

The computational Diffie-Hellman (CDH) problem which the proposed protocol is based on is to find $g^{ab} \pmod q$ when g^a and $g^b \pmod q$ are given for arbitrary $a, b \in Z_q^*$ and generator element g [17]. In other words, when the values of $x_i (= g^{r_i s})$ and $z (= g^s)$ are known in the proposed protocol, a group member can use its own random secret value r_i to calculate $z^{r_i} (= x_i)$ that can be used to recover the session key value, but the probability of a nonmember finding out the value of x_i within a polynomial time is negligibly small, which implies the strong security of the proposed protocol.

In addition, when a new member joins or an original member leaves a group communication, the KDC (or base station) updates its secret key value every session so that a new session key could not be found from the previous group key, leading to the forward and backward secrecies.

V. CONCLUSION

The proposed protocol is expected to be suitable for mobile environment with frequent mobility since it is a group key agreement scheme that based on the computational Diffie-Hellman assumption, only the group members can use the

random secret value of its own selection and the information transmitted from the KDC (or the base station) to recover the common session key by consensus.

Compared with the previous study [7] which extended the TDGH based on the bilinear Diffie-Hellman (BDH) problem, we have used exponential operation that is 3 times faster than pairing operation. Compared with another previous study [8], we have used the hash function and exclusive or operations to reduce the computational cost. Finally, while the security of the EHBT protocol depends on the one-way hash function, the proposed protocol is safe against passive attackers based on the CDH assumption.

Further researches will focus on the efficient management of key tree and the proof of security.

REFERENCES

[1] B. Bhargava, M. Annamalai, and E. Pitoura, "Digital library services in mobile computing", *ACM SIGMOD Record*, vol. 24, no. 4, pp. 34-39, 1995.
[2] Y. Huang and H. Garcia-Molina, "Publish/subscribe in a mobile environment", in *Proceedings MobiDE*, pp. 27-34, 2001.
[3] T. Phan, L. Huang, and C. Dulan, "Challenge: Integrating mobile wireless devices into the computational grid", in *Proceedings MOBICOM*, pp. 271-278, Sep. 2002.
[4] S.-H. Lim and J.-H. Kim, "Real-time broadcast algorithm for mobile computing", *The Journal of Systems and Software*, vol. 69, no. 2, pp. 173-181, 2004.
[5] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644-654, 1976.
[6] Yongdae Kim, Adrian Perrig, and Gene Tsudik, "Group key agreement efficient in communication", *IEEE Transactions on Computers*, vol. 53, no. 7, pp. 905-921, 2004.
[7] Sangwon Lee, Yongdae Kim, Kwangjo Kim, and Dae-Hyun Ryu, "An efficient tree-based group key agreement using bilinear map", in *Proceedings ACNS*, LNCS 2846, pp. 357-371, 2003.
[8] Sandro Rafaeli, Laurent Marthy, and David Hutchison, "EHBT: An efficient protocol for group key management", in *Proceedings NGC*, LNCS 2233, pp. 159-171, Oct. 2001.
[9] Lijun Liao and Mark Manulis, "Tree-based group key agreement framework for mobile ad-hoc networks", Elsevier, *Future Generation Computer Systems*, vol. 23, no. 6, pp. 787-803, 2007.
[10] Sang-won Lee, Jung Hee Cheon, and Yongdae Kim, "Tree-based group key agreement protocol using pairing", *Journal of The Korea Institute of Information Security and Cryptology*, vol. 13, no. 3, pp. 101-110, 2003.
[11] Abhimanyu Kumar and Sachin Tripathi, "Ternary tree based group key agreement protocol over elliptic curve for dynamic group", *International Journal of Computer Applications (0975-888)*, vol. 86, no. 7, pp. 17-25, 2014.
[12] Yvo Desmedt, Tanja Lange, and Mike Burmester, "Scalable authenticated tree based group key exchange for ad-hoc groups", in *Proceedings FC and USEC*, LNCS 4886, pp. 104-118, 2007.
[13] Junghyun Nam, Juryon Paik, Youngsook Lee, Jin Kwak, Ung Mo Kim, and Dongho Won, "Infringing key authentication of an ID-based group key exchange protocol using binary key trees", in *Proceedings KES /WIRN*, Part I, LNAI 4692, pp. 672-679, 2007.
[14] Minghui Zheng, Guohua Cui, Muxiang Yang, and Jun Li, "Scalable group key management protocol based on key material transmitting tree", in *Proceedings ISPEC*, LNCS 4464, pp. 301-313, 2007.
[15] K. Becker and U. Wille, "Communication complexity of group key distribution", in *Proceedings CCS*, pp. 1-6, 1998.
[16] P.S.L.M. Barreto, H.Y. Kim, B. Linn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems", in *Proceedings Advances in Cryptology-Crypto*, LNCS 2442, pp. 354-368, 2002.
[17] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp. 113, 1997.