Secure Group Communication with Dynamic Membership Change in Ad Hoc Networks

Heeyoul Kim
Department of Computer Science, Kyonggi University
Republic of Korea
[e-mail: heeyoul.kim@kgu.ac.kr]

Received May 3, 2011; revised July 7, 2011; August 2, 2011; accepted September 18, 2011; published September 29, 2011

Abstract

The importance of secure communication between only legitimate group members in ad hoc networks has been growing in recent years. Due to the ad hoc nature the scalability on dynamic membership change is a major concern. However, the previous models require at least $O(\log n)$ communication cost for key update per each membership change, which imposes a heavy burden on the devices. In this paper we present a scalable model that supports communication-efficient membership change in ad hoc networks by exclusionary keys and RSA functions. The multicast cost for key update is extremely low, that is $O(1)$, and one-to-one communications occur mostly in neighboring devices.

Keywords: Secure group communication, group key rekeying, ad hoc network

This work was supported by Kyonggi University Research Grant 2009.

DOI: 10.3837/tiis.2011.09.010

1. Introduction

A mobile ad hoc network is a self-configuring network of communication devices which carry out networking functions such as packet forwarding, routing, and network management via wireless connection. In ad hoc networks, security is a crucial issue [1] since communications are exposed in the wireless environments and the devices are vulnerable to be stolen or lost.

As group communication in ad hoc networks has become one of the most emerging fields and has been widely investigated, the importance of secure group communication model has also been recognized, especially in the applications such as military communications in battlefield or commercial communications where security is essential. The major objective of secure group communication is to provide a secure and reliable way to protect communications among group members from other entities. A cryptographic solution is to encrypt transferred messages with a group key which is a secret key only known to the group members.

In many group-based applications, the membership is changed dynamically: a new member may join the group some time later and an existing member may leave the group. This change causes the group key to be updated to prevent a new member from discovering previous messages and to prevent a leaving member from discovering future messages. Group key update operation is complex and consumptive because the group size may be very large. Hence scalability is the most important issue, and especially the dominant cost is rather communication cost than computation cost [2][3].

A lot of group key distribution schemes for secure group communication have been presented in the wired network, e.g. LKH [4][5], OFT [6], and ELK [7]. Among them the most representative scheme is Logical Key Hierarchy (LKH) scheme which uses a logical key tree for group key update. Another approach is group key agreement schemes including GDH [8] and TGDH [9] where all members equally participate in agreeing a group key through the extension of Diffie-Hellman key exchange algorithm.

Considering the distributed nature of ad hoc and wireless sensor networks, a number of secure group communication models have been proposed in the literature. Lazos and Poovendran [10][11] proposed two secure group communication models that construct a hierarchical structure according to group members' location information with the assistance of GPS module. TKH [12] generates a key tree by using the underlying network topology for energy efficiency. Pietro et al. [13] merges the extension of LKH with directed diffusion. The models in [14][15][16] extend traditional group key agreement schemes for ad hoc networks.

Although previous models strive to reflect the characteristics of ad hoc networks, there still remains a limitation that at least $O(\log n)$ size message is multicasted to all group members per each key update. The cost of multicasting in ad hoc networks is larger than the one in wired network because each group member has to get the assistance of other devices. Moreover, in ad hoc networks group membership is changed very dynamically because the devices have mobility and the links between devices are unreliable. Therefore this limitation imposes a heavy burden on the devices that have a restricted battery power.

In this paper, we present a fully distributed and very scalable model for secure group communication in ad hoc networks. First, we present the concept of exclusionary key that is securely shared with all members excluding some subset of them. Then we propose a scalable model which reduces communication cost using exclusionary keys with RSA functions.

Especially, our model is focused on reducing the size of multicast message from $O(\log n)$ to $O(1)$ per each key update. Moreover, it enables the number of hops in one-to-one communication to be very small by performing only local communications.

The rest of the paper is organized as follows. In Section 2, we describe the requirements for secure group communication in ad hoc networks. In Section 3, we explain the tree structure and exclusionary keys used in our model. Section 4 provides a scalable and secure group communication model and Section 5 analyzes our model in the aspect of both security and performance. In Section 6 we discuss several issues including synchronization, and Section 7 concludes the paper.

2. Requirements for Secure Group Communication in Ad Hoc Networks

In this section, we explain the following requirements which secure group communication in ad hoc networks should satisfy. Among them, the first three requirements consider the characteristics of ad hoc networks and the others consider the security of group communication.

- Distributed Model
  The management of group key and group membership can be centralized or distributed. Centralized management assumes the existence of a center which manages the group and distributes keys [17][18]. Distributed management refers to managing keys and membership between the group members without outside assistance [9][19][20]. Because in ad hoc networks the devices are distributed and there is no pre-existing infrastructure, the latter is more suitable.
- Efficiency
  Because the devices composing an ad hoc network usually have a low power and thus power saving is a major concern, the efficiency of group management is one of the main requirements. Especially key update operation must be scalable even if the group size is very large. When a new member joins or an existing member leaves the group, the cost to be performed by the other members must be minimized as possible.
- Reliability
  Because the links between devices are unreliable, the group may be partitioned into small groups. Even in this case, communication between the members in a small group should be enabled. If a specific member manages the whole group, most of the small groups are beyond the scope of the member. On the other hand, if all members have the same role, the small groups can be preserved individually.
- Group Key Secrecy
  It guarantees that it is computationally infeasible for an adversary outside the group to discover the group key. This is a basic requirement in secure group communication.
- Forward Secrecy
  It guarantees that an adversary having a number of previous keys cannot discover current group key. It implies that even an outside adversary who was a member of the group in the past, but is not now cannot discover current messages.
- Backward Secrecy
  It guarantees that an adversary having current key cannot discover previous group keys. It implies that even a new member joining the group currently cannot discover any of previous messages.

3. Tree Structure with Exclusionary Keys

Let us assume that there is a logical binary key tree where each node represents a key and each leaf node corresponds to each member. Let n_1 be the root node and each node is assigned a binary index which is appending 0 to the index of parent node if the node is a left child, or appending 1 if the node is a right child. For example, the index of a left child node of n_1 becomes n_{10} and the index of a right child node becomes n_{11}. Here we define an exclusionary key EK as follows.

Definition 1: An exclusionary key EK_i of a node n_i is a secret key shared with all members except the members in the subtree rooted at the node n_i.

To reduce the number of stored keys, each exclusionary key except EK_1 is derived from the parent key by the RSA functions rather than generated independently. An RSA function $f : \mathbb{Z}_N \rightarrow \mathbb{Z}_N$ is defined as

$$f(x) = x^e \bmod N, \tag{1}$$

where $N = pq$, p and q are large distinct primes, $gcd(e,(p-1)(q-1)) = 1$ and $|N| = l$. The values (N,e) are public, and the values (p,q) are secret. The function f has both one-wayness property and multiplicative property explained below.

- **One-wayness Property**
  It is easy to compute $f(x)$ for all $x \in \mathbb{Z}_N$, but it is computationally infeasible to find any x such that $f(x) = y$ for a given $y \in \mathbb{Z}_N$.
- **Multiplicative Property**
  For all $x, y \in \mathbb{Z}_N$, the following is satisfied:

$$\begin{aligned} f(x)f(y) &= (x^e \bmod N)(y^e \bmod N) \\ &= (xy)^e \bmod N \\ &= f(xy) \end{aligned} \tag{2}$$

The former property prevents a user from discovering prohibited keys and the latter property enables to update exclusionary keys efficiently, which will be explained in Section 4.

Let f_L and f_R be two instances of the RSA function with different parameters such that

$$f_L(x) = x^{e_L} \bmod N, \quad f_R(x) = x^{e_R} \bmod N \tag{3}$$

where $N = pq$, and $|N| = l$. After EK_1 is randomly generated, other keys are derived from EK_1 by the following equation:

$$EK_{i\|0} = f_L(EK_i), \quad EK_{i\|1} = f_R(EK_i). \tag{4}$$

In other words, the exclusionary key of a node can be computed from the key of parent node with the function f_L if it is a left child, or f_R otherwise.

Then each member u corresponding to a leaf node n_i gets and stores only the exclusionary keys of sibling nodes of the nodes along the path from root to n_i, which are $\log n$ keys where n is the number of group members. We call the set of keys $SibSet_i$. With the keys in $SibSet_i$, he can compute the key of any leaf node except EK_i using **Algorithm 1**. For example, a member u_2 in **Fig. 1** is able to compute $EK_{1000} \sim EK_{1111}$ except EK_{1001} because he knows $SibSet_{1001} = \{EK_{11}, EK_{101}, EK_{1000}\}$. On the other hand, he cannot compute the keys EK_1, EK_{10}, EK_{100} and EK_{1001} by the one-wayness property.

Algorithm 1. Exclusionary key derivation

```
Input  : $SibSet$, index $i = b_1 b_2 ... b_t$ where $b_j = 0$ or 1, $1 \le j \le t$
Output : $EK_i$
  Find a key $EK_k \in SibSet$ where its index $k$ matches the prefix of $i$
  $K \leftarrow EK_k$
  $t' \leftarrow |k|$                    // $t'$ : length of $EK_k$'s index
  for $j = t' + 1$ to $t$ do
    if $b_j = 0$ then
      $K \leftarrow f_L(K)$
    else
      $K \leftarrow f_R(K)$
    end if
  end for
  return $K$
```

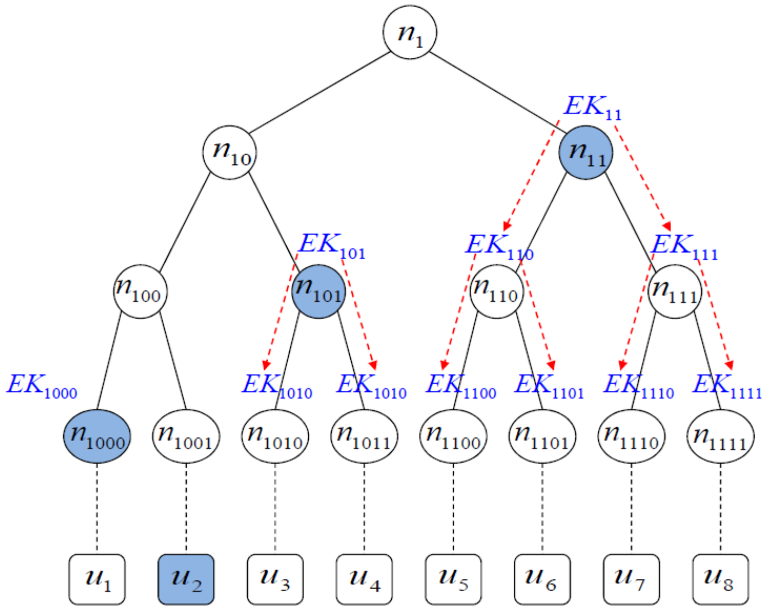

Fig. 1. Exclusionary key computation of a user u_2

In this section, we present a scalable and secure group communication model for ad hoc networks. Its main advantage is that very few messages are transferred to the members for key update. Especially, the size of multicast message transferred to all group members for key update is very small.

4.1 Assumptions

Before two devices communicate with each other, each device authenticates the other side with the public key certificates. Due to the distributed nature of ad hoc networks, we cannot assume that there exists a trusted CA issuing certificates by binding a public key to a device's identity. Instead, a distributed public key management service using threshold cryptography [21] and a self-organized public key infrastructure [22] have been presented.

After authentication, two communicating devices establish a secure one-to-one channel by the Diffie-Hellman key change protocol with their private/public key pairs. This channel provides confidentiality of transferred information.

4. Secure Group Communication Model

Each device u has a unique identifier uid_u which is specified in his certificate. The identifier is confirmed by the authentication process, and it provides the way to check whether he is already the group member or not in Section 4.4.

We also assume that some initial members are known before the network is constructed. Though predicting the whole potential group members is impossible, predicting a small subset of them is possible in many cases and initializing with them has a better performance. In an extreme case, setup is done with just two initial members and other members are added afterward by join protocol.

4.2 Setup

Before the devices participate in the ad hoc network, an off-line server sets up security parameters l and m. l means the security level of RSA functions and m means the security level of symmetric encryption scheme.

The server then selects two RSA functions f_L and f_R. The length of exponents e_L and e_R should be chosen carefully because the cost of function computation is very closely dependent on the length of them, but choosing too small value such as 3 makes the proposed model vulnerable. The functions f_L, f_R are very similar to RSA signature verification operation, and it is well known that using a small public exponent reduces operation time. Especially, according to [23] the value $e = 2^{16} + 1 = 65537$ is recommended to endure some attacks such as Coppersmith's theorem [24] or Hastad's broadcast attack [25]. On the performance aspect, when $e = 2^{16} + 1 = 65537$ is used, the function requires only 17 multiplications as opposed to roughly 1000 multiplications when it is randomly chosen. Therefore $2^{16} + 1$ or similar number are recommended for the exponents.

The server also selects two cryptographic secure hash functions:

$$H_1 : \{0,1\}^l \rightarrow \{0,1\}^m, \quad H_2 : \{0,1\}^m \rightarrow \{0,1\}^m.$$

H_1 is utilized to derive an encryption key from an exclusionary key, and H_2 is utilized to update group key.

With initial members, the server constructs a key tree explained in Section 3 and randomly generates an initial group key GK with m bits. The server also randomly generates EK_1 with l bits, and computes the other exclusionary keys with f_L and f_R by the Equation (4).

Each member u receives the following values from the server and stores them.

- ID_u : index of the node corresponding to u in the key tree
- $SibSet_{ID_u}$: a set of the exclusionary keys of sibling nodes of the nodes along the path from the root to n_{ID_u}
- GK : initial group key

4.3 Group Communication

When a group member wants to send a message to other members, the message must be transferred securely. A symmetric key encryption scheme such as AES with current group key GK provides confidentiality of the message because only current group members know the key.

4.4 Join Protocol

If a new device joins the group, current group key should be updated to prevent him from discovering previous messages. Because the join request occurs frequently in ad hoc networks, it is necessary to minimize the costs of other members as possible. Compared with the previous models, most of the communications in this protocol are achieved in a local area.

Let's assume that a new device u wants to join the group. He finds two nearest group members with a range-limited broadcast using a TTL bound. We call the first member a *sponsor* S and call the second member a *cooperator* C. The steps of join protocol are as follows.

1. $u \to sponsor$
   A. u sends a join request to S.
   B. From this request, the receiver is appointed to be a sponsor.
2. $sponsor \to all$
   A. After proper inspection, S floods a join approval message with uid_u.
   B. Then each member derives a new group key $GK' = H_2(GK)$. Only the group members can derive the next group key.
3. $sponsor \to u$
   A. S assigns $ID_u = ID_S \| 1$. Then S sends ID_u, GK' and $SibSet_{ID_S}$ to u.
   B. The index of S is changed to $ID_S = ID_S \| 0$.
4. $sponsor \to cooperator$
   A. S sends C a join cooperation request including ID_S and ID_u.
5. $cooperator \to u$
   A. C computes EK_{ID_S} by the Algorithm 1 and sends it to u. Now, u holds all the keys in $SibSet_{ID_u}$.
6. $cooperator \to sponsor$

A. C also computes EK_{ID_u} by the Algorithm 1 and sends it to S. Then S holds all the keys in $SibSet_{ID_S}$.

For example, suppose that u_9 wants to join the group in **Fig. 2** and the two nearest members are u_4 and u_8. Then u_4 becomes a sponsor and u_8 becomes a cooperator. After receiving join request, u_4 sends $ID_{u_9} = 10111$, $SibSet_{1011} = \{EK_{11}, EK_{100}, EK_{1010}\}$ to u_9 and ID_{u_4} is changed to 10110. Then u_8 computes EK_{10110} from $EK_{10} \in SibSet_{1111}$ and sends it to u_9. Now u_9 holds all the keys in $SibSet_{10111} = \{EK_{11}, EK_{100}, EK_{1010}, EK_{10110}\}$. On the other hand, u_4 receives EK_{10111} from u_8 which completes $SibSet_{10110}$.

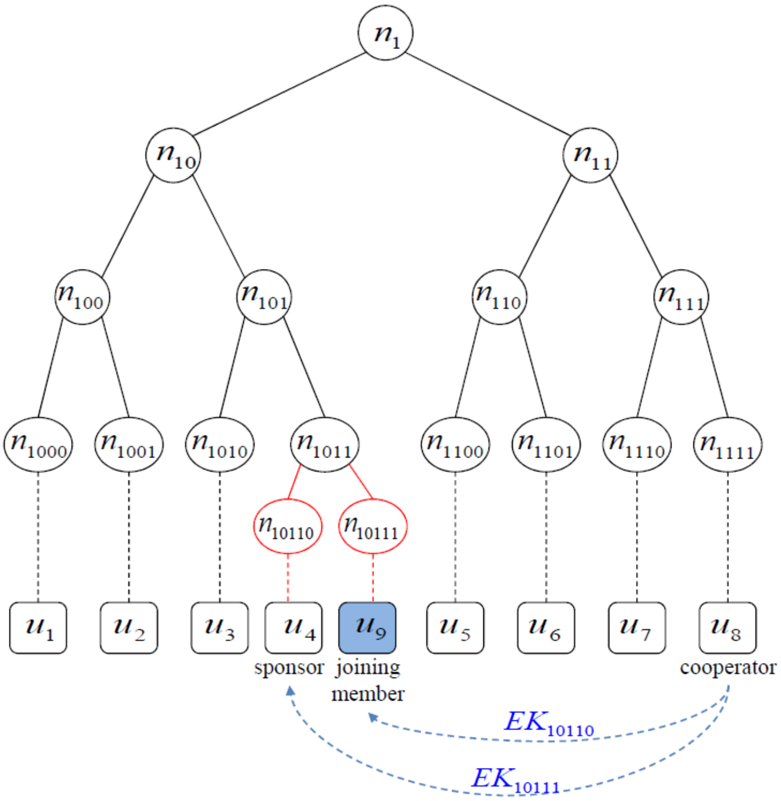

Fig. 2. A key tree after u_9 joins the group

4.5 Leave Protocol

A leave request of an existing member also occurs frequently in ad hoc networks. In this case, the keys that leaving member knows should be updated to prevent him from overhearing the communicated messages in the future.

Let's assume that a member u wants to leave the group. He finds the nearest group member with a range-limited broadcast. We call the member a *revoker* R. Then the following steps are performed.

1. $u \to revoker$
   A. u sends R a leave request with his index ID_u.
   B. After receiving the request, R randomly generates a new group key GK' and an update value U_1.
2. $revoker \to all$
   A. R computes EK_{ID_u} from $SibSet_{ID_R}$ and derives an encryption key $H_1(EK_{ID_u})$.
   B. GK' and U_1 are encrypted with $H_1(EK_{ID_u})$ and multicasted to all members. Also ID_u and uid_u are multicasted.
   C. Each member computes EK_{ID_u}, and obtains GK' and U_1 by decrypting the message. Only the members in the group except u can decrypt the message as described in Section 3.

With the update value U_1, each member updates his exclusionary keys by **Algorithm 2**. For each exclusionary key EK_i he holds, U_i is computed by the following equation:

$$U_{i\|0} = f_L(U_i), \quad U_{i\|1} = f_R(U_i). \tag{5}$$

Then EK_i' is computed by multiplying EK_i by U_i. For example, as can be seen in **Fig. 3**, a member u_2's next keys are computed as

$$\begin{aligned} EK_{11}' &= EK_{11} \cdot f_R(U_1) \bmod N \\ EK_{101}' &= EK_{101} \cdot f_R(f_L(U_1)) \bmod N \\ EK_{1000}' &= EK_{1000} \cdot f_L(f_L(f_L(U_1))) \bmod N \end{aligned} \tag{6}$$

Because both f_L and f_R have multiplicative property, updated exclusionary keys preserve the following relation and the **Algorithm 1** is kept correctly.

$$EK_{i\|0}' = EK_{i\|0} \cdot U_{i\|0} = f_L(EK_i) \cdot f_L(U_i) = f_L(EK_i \cdot U_i) = f_L(EK_i') \tag{7}$$

$$EK_{i\|1}' = EK_{i\|1} \cdot U_{i\|1} = f_R(EK_i) \cdot f_R(U_i) = f_R(EK_i \cdot U_i) = f_R(EK_i') \tag{8}$$

If an existing member u' finds out that u was the sibling member of u in the key tree, in other words $ID_{u'}$ is equal to ID_u except the last one bit, his index is right shifted and his logical position is changed to corresponding parent node.

Algorithm 2. Exclusionary key update

```
Input   : $SibSet_i$, $U_1$
Output : $SibSet_i$
  $ind \leftarrow 1$
  $U \leftarrow U_1$
  $d \leftarrow |SibSet_i|$          // $d$ : number of keys in $SibSet_i$
  for  $j = 1$ to $d$  do
```

```
        if $EK_{ind\|0} \in SibSet_i$ then
            $U' \leftarrow f_L(U)$                    $// U' = U_{ind\|0}$
            $EK_{ind\|0} \leftarrow EK_{ind\|0} \cdot U' \bmod N$
            $U \leftarrow f_R(U)$                     $// U = U_{ind\|1}$
            $ind \leftarrow ind \| 1$
        else                                          $// EK_{ind\|1} \in SibSet_i$
            $U' \leftarrow f_R(U)$                    $// U' = U_{ind\|1}$
            $EK_{ind\|1} \leftarrow EK_{ind\|1} \cdot U' \bmod N$
            $U \leftarrow f_L(U)$                     $// U = U_{ind\|0}$
            $ind \leftarrow ind \| 0$
        end if
    end for
    return $SibSet_i$
```

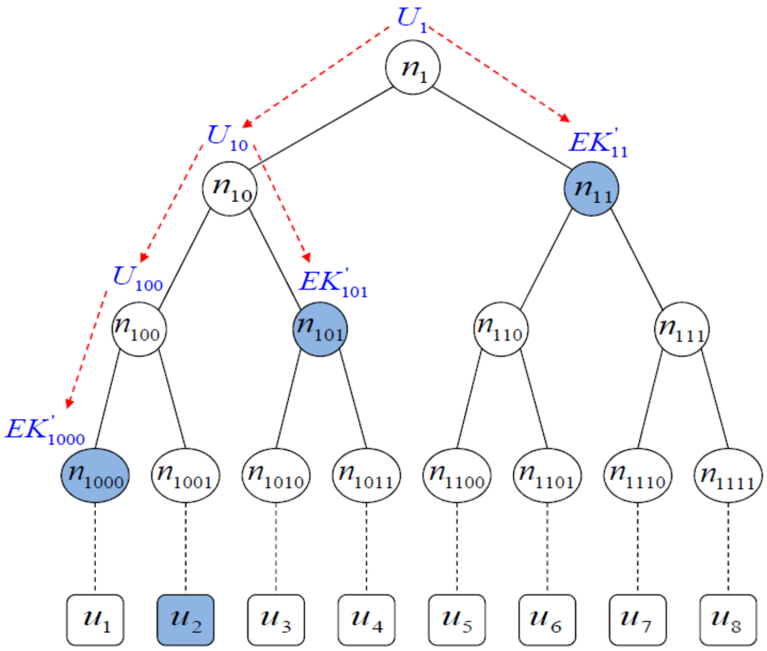

Fig. 3. Exclusionary key update of a user u_2

5. Analysis of Proposed Model

5.1 Performance

Because ad hoc devices usually have a low power and a small computation capacity, performance is one of the most important factors. The major measurements of performance evaluation are storage cost, computation cost and communication cost. Here we will analyze the presented model and compare it with previous models. The costs of each device are very closely dependent on the depth from the root to the corresponding leaf node in the key tree. Let n be the number of group members. Then the depth is roughly equal to $\log n$.

From the viewpoint of storage cost, each member u stores his index ID_u, a group key GK, and a key set $SibSet_{ID_u}$ which is made up of $\log n$ exclusionary keys. However, he does not need to store any other information about the key tree structure.

Table 1 and 2 show the result of comparison with previous distributed secure group communication models. Among them, BD [26], GDH [8] and TGDH [9] are key agreement approaches and the others are key distribution approaches.

Table 1. The number of transferred keys

model	join		leave	
	multicast	**unicast**	**multicast**	**unicast**
BD [26]	$2n$	0	$2n$	0
GDH [8]	n	$n-1$	n	$\log n-1$
TGDH [9]	$\log n$	$\log n+1$	$\log n$	0
D-LKH [27]	$2\log n$	$\log n+1$	$2\log n-1$	0
D-OFT [28]	$\log n$	$\log n+1$	$\log n$	0
Proposed	0	$\log n+2$	2	0

Table 2. Computation cost comparison of a member

model	join	leave
BD [26]	$(n+1)^{*}$	$(n+1)^{*}$
GDH [8]	$(n+1)^{*}$	$(n+1)^{*}$
TGDH [9]	$\log n^{*}$	$\log n^{*}$
D-LKH [27]	$(\log n+1)^{\&}$	$\log n^{\&}$
D-OFT [28]	$(\log n+1)^{\&} + \log n^{\wedge}$	$1^{\&} + \log n^{\wedge}$
Proposed	$1^{\wedge} + (\log n+1)^{\%\#}$	$2^{\&} + 1^{\wedge} + (3\log n)^{\%}$

*modular exponentiation, $^{\&}$decryption, $^{\wedge}$hash function
$^{\%}$RSA function, $^{\#}$only done by cooperator

In join protocol, a new member u receives his index, a group key and $\log n+1$ exclusionary keys from the sponsor and the cooperator. The cost of transferring them is much smaller than a general one-to-one communication cost because u communicates with the most nearest two members. Moreover, the size of multicast messages sent from the sponsor to all members are very short. The cooperator computes two descendent exclusionary keys from his $SibSet$, and sends one key to u and the other key to the sponsor. Because the keys can be computed simultaneously, at most $\log n+1$ RSA function computations are done by the cooperator.

In leave protocol, the revoker multicasts an encrypted message including GK' and U_1.

Compared with previous models which multicast a message including $\log n$ keys at least, the communication cost is much little. After receiving the encrypted message, each member computes the encryption key with at most $\log n$ RSA function computations. Then he updates his exclusionary keys with at most $2\log n$ function computations.

In **Table 1**, proposed model significantly reduces the size of multicast message to $O(1)$ in both join and leave protocol with a slight increase of unicast message size. Because communication cost is the dominant cost as we pointed out, such advantage of proposed model is very meaningful. In **Table 2**, computing an RSA function is much faster than computing a general modular exponentiation because the exponents e_L and e_R in RSA functions can be very small. Thus the proposed model has a better performance than key agreement approaches. Though the computation cost in leave protocol exceeds the cost of D-LKH and D-OFT, each members except the cooperator has a little computation cost in join protocol.

Because the logical position of a new member in the tree is dependent on the physical location of it, the depth of a member becomes deeper than $\log n$ after many members join the group. However, it is very costly to select an ideal position that minimizes the maximum depth of the tree because it requires a global information including tree structure and all members' indexes, and may need a communication between very distant members. On the other hands, we argue that there is little difference between the depth in the presented model and $\log n$ because the devices have mobility and a join request generally occurs in a random place. We estimated the actual tree depth by simulation. In the simulation, initially the group is made up of 1000 devices, and 1000 join requests and 1000 leave requests occurs in random places by turns.

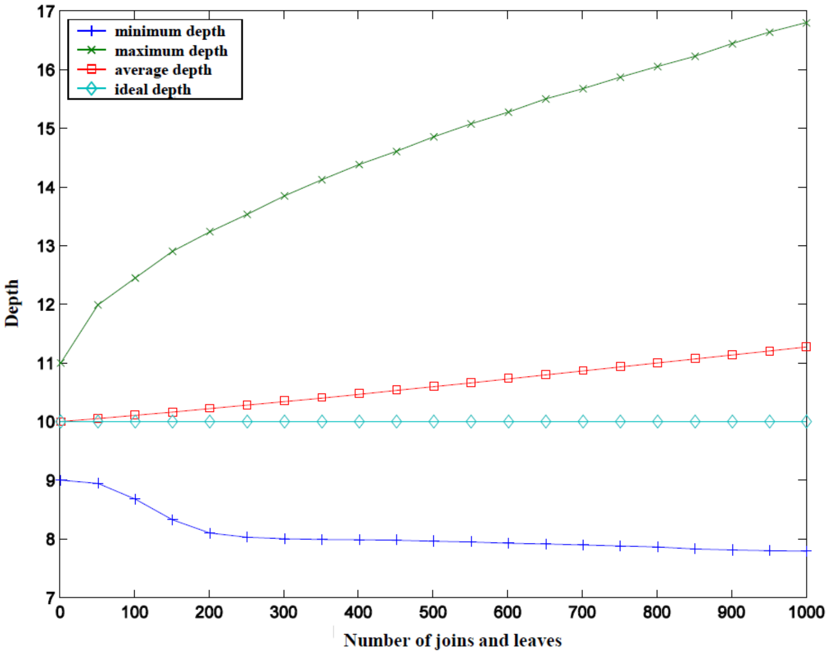

Fig. 4. Increase of depth according to consecutive join and leave

Fig. 4 shows the increase of tree depth, and **Fig. 5** shows the distribution of members according to depth. Although the maximum depth increases roughly twofold compared to an ideal depth $\log n$, just only one depth is additionally raised on the average. Moreover, almost members have the depth adjacent to $\log n$. It means that total costs of the whole network increases just a little compared to ideal costs.

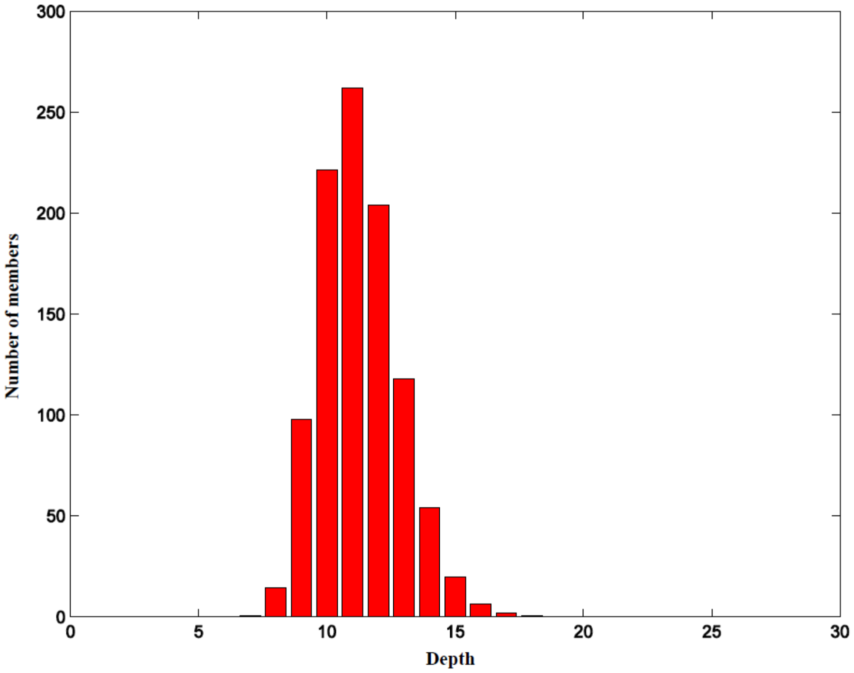

Fig. 5. Distribution of members according to depth after 1000 joins and leaves

5.2 Security

The security of our model basically depends on the difficulty of finding inverse of RSA function, and it is widely believed to be computationally equivalent to factoring large integer. Based on it, we briefly analyze our model with major security requirements in secure group communication that are referenced from [8][29]. In this analysis, we assume that an adversary is outside the group even though he might be in the group in the past.

- Group Key Secrecy
  Suppose that an adversary is out of the group, and tries to discover current group key. In our model, initial group key is only distributed to initial members with a secure channel. Whenever a new member joins, a new group key is derived from the previous key and securely sent to the new member. So the new key is shared with only current group members. Thus, the adversary needs to perform $\Omega(2^m)$ operations to brute-force a group key with overwhelming probability. Whenever an existing member leaves, a new group key is chosen independently and encrypted with the key derived from some exclusionary key that only the remaining members holds. So the adversary also needs to perform $\Omega(2^m)$ operations to brute-force the encryption key with overwhelming probability, or needs to find inverse of RSA function to discover the exclusionary key.
- Forward Secrecy
  Suppose that an adversary was in the group with holding some previous keys, and tries to discover current group key. When he leaves the group, he might overhear the encrypted message including a new group key and an update value. However, he cannot decrypt the

message because it is encrypted with the key he cannot discover by the definition of exclusionary keys. After key update, his previous keys cannot be any help to him. Therefore the adversary needs to perform $\Omega(2^m)$ operations to brute-force the encryption key or factorize the large composite number N.

- Backward Secrecy
  Suppose that an adversary joins the group, and tries to discover previous group key. He obtains a new group key and some exclusionary keys. However, because of one-wayness property of hash function, he cannot guess previous group key just before he joins. Therefore the adversary needs to perform $\Omega(2^m)$ operations to brute-force the previous group key with overwhelming probability. He might overhear the encrypted message including the update value and a new group key during the last leave protocol before he joins the group. However, with his current exclusionary keys he cannot discover the encryption key derived from one of the previous exclusionary keys because he does not know the update value. Therefore the adversary also needs to perform $\Omega(2^m)$ operations to brute-force the encryption key with overwhelming probability.

6. Discussion

Synchronization between group members is another well-known problem in secure group communication [30]. During key update, update message may be missed by a member if the link is broken or he exhausts his battery power. Then he is inadvertently excluded from receiving group communications. Though this problem has been studied in the context of reliable multicast [31][32], in ad hoc networks it becomes more difficult because there is no infrastructure that controls global synchronization. A trivial solution is that the out-of-synch member re-joins the group, but it requires additional costs. Another way is to use a sequence number to check consistency and re-transmit update message if the number is invalid. This way also requires additional communication cost. However, it can be done more efficiently in our model because the size of update message is very small.

The broadcast encryption, which has been studied in a different way [33][34], has a similar object that only selected members can receive secret messages. Naor et al. [34] proposed a model, called by subset difference, in which the size of multicast message is $O(r)$ where r is the number of totally revoked members. However it is not applicable to group communication in ad hoc networks because it assumes that group membership is fixed prior to network construction and only one member can send message to others. Moreover, it is relatively inefficient for dynamic membership change because r increases continuously as members leave the group [35].

7. Conclusion and Future Work

In this paper we have proposed a secure group communication model that is suitable for ad hoc networks. Because our model is fully distributed, it requires no centralized server to manage the group, and it tolerates a single point of failure. With the concept of exclusionary keys and RSA functions, our model requires extremely low multicast cost $O(1)$ keeping computation cost similar to the previous key distribution models: in the join protocol only a small-sized message is multicasted, and in the leave protocol just two encrypted keys are multicasted. Moreover, one-to-one communication occurs in a local area in the majority of time. These advantages enable the proposed model to be very scalable and efficient in ad hoc network environments.

Another approach for improving tree-based model is considering network topology or location information when constructing tree hierarchy, as mentioned in Section 1. We think this approach may be applied to our model for improvement. We plan to further investigate how to use network topology when creating and managing key tree with exclusionary keys.

References

[1] R. Molva, P. Michiardi, “Security in Ad hoc Networks,” in *Proc. of Personal Wireless Communications (PWC'03)*, pp. 736-775, 2003. Article (CrossRef Link)

[2] Y. Amir, Y. Kim, C. Nita-Rotaru, G. Tsudik, “On the Performance of Group Key Agreement Protocols,” in *Proc.of 22nd IEEE International Conference on Distributed Computing Systems*, pp. 463-464, 2002. Article (CrossRef Link)

[3] Y. Kim, A. Perrig, G. Tsudik, “Communication-efficient group key agreement,” in *Proc. of the 16th International Conference on Information Security: Trusted Information*, pp. 229-244, 2001.

[4] C.K. Wong, M.G. Gouda, S.S. Lam, “Secure Group Communications using Key Graphs,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, 2000. Article (CrossRef Link)

[5] D. Wallner, E. Harder, R. Agee, “Key Management for Multicast: Issues and Architectures,” RFC 2627, 1999.

[6] D.A. McGrew, A.T. Sherman, “Key Establishment in Large Dynamic Groups using One-way Function Trees,” *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444-458, 2003. Article (CrossRef Link)

[7] A. Perrig, D. Song, D. Tygar, “ELK: A New Protocol for Efficient Large Group Key Distribution,” in *Proc. of the IEEE Security and Privacy*, 2001. Article (CrossRef Link)

[8] M. Steiner, G. Tsudik, M. Waidner, “Key Agreement in Dynamic Peer Groups,” *IEEE Trans. on Parelled Distributed Systems*, vol. 11, no. 8, pp. 769-780, 2000. Article (CrossRef Link)

[9] Y. Kim, A. Perrig, G. Tsudik, “Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups,” *ACM Conference on Computer and Communications Security*, pp. 235-244, 2000. Article (CrossRef Link)

[10] L. Lazos, R. Poovendran, “Energy-aware Secure Multicast Group Communication in Mobile Networks,” in *Proc. of IEEE International conference on Acoustics Speech and Signal Processing*, pp. 201-204, 2003.

[11] L. Lazos, R. Poovendran, “Location-aware Secure Wireless Multicast in Ad Hoc Networks under Heterogeneous Pathloss,” in *Proc. of UWEETR-2003-0012*, 2003.

[12] J. Son, J. Lee, S. Seo, “Topological Key Hierarchy for Energy-Efficient Group Key Management in Wireless Sensor Networks,” *Wireless Personal Communications*, vol. 52, pp. 359-382, 2010. Article (CrossRef Link)

[13] R.D. Pietro, L.V. Mancini, Y.W. Law, S. Etalle, P.J.M. Havinga, “LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks,” in *Proc. of ICPP Workshops*, 2003. Article (CrossRef Link)

[14] N. Asokan, P. Ginzboorg, “Key-Agreement in Ad-hoc Networks,” *Computer Communications*, vol. 23, no. 17, pp. 1627-1637, 2000. Article (CrossRef Link)

[15] T. Chiang, Y. Huang, “Group Keys and the Multicast Security in Ad Hoc Networks,” in *Proc. of ICPP Workshops*, pp. 385-390, 2003. Article (CrossRef Link)

[16] Y. Wang, X.Y. Li, O. Frieder, “Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Networks,” in *Proc. of the 11th International Conference on Computer Communications and Networks(ICCCN'02)*, pp. 147-151, 2002. Article (CrossRef Link)

[17] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, D. Saha, “Key Management for Secure Internet Multicast using Boolean Function Minimization Techniques,” in *Proc. of IEEE Infocomm'99*, pp. 689-698, 1999. Article (CrossRef Link)

[18] R. Safavi-Naini, H. Wang, “New Constructions for Multicast Re-keying Schemes using Perfect

Hash Families," in *Proc. of the 7th ACM conference on Computer and communications security*, pp. 228-234, 2000. Article (CrossRef Link)

[19] I. Ingemarsson, D. Tang, C. Wong, "A Conference Key Distribution system," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 714-720, 1982. Article (CrossRef Link)

[20] A. Perrig, "Efficient Collaborative Key Management Protocols for Secure Autonomous Group Communication," in *Proc. of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pp. 192-202, 1999.

[21] L. Zhou, Z. Haas, "Securing Ad Hoc Networks," *IEEE Networks Magazine*, vol. 13, no. 6, pp. 24-30, 1999. Article (CrossRef Link)

[22] S. Capkun, L. Buttyan, J.P Hubaux, "Self-Organized Public Key Management for Mobile Ad Hoc Networks," *ACM International Workshop on Wireless Security*, 2002. Article (CrossRef Link)

[23] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," *Notices of the American Mathematical Society (AMS)*, vol. 46, No. 2, pp. 203-213, 1999.

[24] D. Coppersmith, "Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities," *Journal of Cryptology*, vol. 10, pp. 233-260, 1997. Article (CrossRef Link)

[25] J. Hastad, "Solving Simultaneous Modular Equations of Low Degree," *SIAM Journal of Computing*, vol. 17, pp. 336-341, 1988. Article (CrossRef Link)

[26] M. Burmester, Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," in *Proc. of Advances in Cryptology, Eurocrypt '94*, pp. 275-286, 1994. Article (CorssRef Link)

[27] O. Rodeh, K. Birman, D. Dolev, "The Architecture and Performance of Security Protocols in the Ensemble Group Communication System: Using Diamonds to Guard the Castle", *ACM Transactions on Information System Security*, vol. 4, no. 3, pp. 289-319, 2001. Article (CrossRef Link)

[28] L. Dondeti, S. Mukherjee, A. Samal, "DISEC: A Distributed Framework for Scalable Secure Many-to-Many Communication," in *Proc. of the Fifth IEEE Symposium on Computers and Communications*, pp. 693-698, 2000. Article (CrossRef Link)

[29] Y. Kim, A. Perrig, G. Tsudik, "Tree-based Group Key Agreement," *ACM Transactions on Information System Security*, vol. 7, no. 1, pp. 60-96, 2004. Article (CrossRef Link)

[30] M. Baugher, R. Canetti, L. Dondeti, F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture," RFC 4046, 2005.

[31] B. Adamon, C. Bormann, M. Handley, J. Macker, "Negative-Acknowledgment (NACK) -Oriented Reliable Multicast (NORM) Protocol," RFC 3940, 2004.

[32] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation," RFC 3450, 2002.

[33] A. Fiat, M. Naor, "Broadcast Encryption," in *Proc. of the 13th Annual International Cryptology Conference on Advances in Cryptology*, pp. 480-491, 1994. Article (CrossRef Link)

[34] D. Naor, M. Naor, J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," in *Proc. of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 41-62, 2001. Article (CrossRef Link)

[35] S. Zhu, S. Jajodia, "Scalable Group Key Management for Secure Multicast: a Taxonomy and New Directions," *Network Security*, pp. 57-75, 2010. Article (CrossRef Link)

Heeyoul Kim received the B.E. degree in Computer Science from KAIST, Korea, in 2000, the M.S. degree in Computer Science from KAIST in 2002, and the Ph.D. degree in computer science from KAIST in 2007. From 2007 to 2008, with the Samsung Electronics as a senior engineer. Since 2009 he has been a faculty member of Division of Computer Science at Kyonggi University. His main research interests include cryptography & security such as secure group communication and clod computing security.