SMP: Scalable Multicast Protocol for Granting Authority in Heterogeneous Networks

Kuo-Jui Wei[1], Jung-San Lee[1], and Bo Li[2]
(Corresponding author: Jung-San Lee)

Department of Information Engineering and Computer Science[1]
Feng Chia University, Taichung 40724, Taiwan, R.O.C.
Department of Electrical Engineering and Computer Science[2]
Vanderbilt University, Nashville, Tennessee, USA
(Email: leejs@fcu.edu.tw)
(Received Aug. 6, 2015; revised and accepted Jan. 23 & Feb. 28, 2016)

Abstract

The fundamental function of the network protocol is to provide confidential communications or services for authorized participants over an insecure network. The proliferation of the Internet and mobile computing technologies, however, has led to emerging applications such as message services, pay-per-view, teleconference, and collaboration tasks. Especially, users now can surf over the Internet or get online for communications via wired, wireless, 3G, or LTE (Long Term Evolution) networks. Traditional peer-to-peer transmission protocol will no longer suffice for these types of applications. Consequently, point-to-group and group-to-group transmission have become important areas of focus. The main challenge in designing a secure multicast mechanism results from large groups and frequent key updates caused by members joining and leaving. To mitigate the encumbrance of group high-mobility in heterogeneous networks, we propose a subgroup-based multicast protocol adopting Lagrange Interpolating Polynomial technique. Simulation results show that the scalable multicast protocol (SMP) can not only preserve the forward and backward secrecy of group communications but also perform better than related works on system communication cost and storage consumption.

Keywords: Communication security, multicast, scalability

1 Introduction

Engineers have proposed many security protocols for providing confidential communications in large network groups; protocols for multicast communications are regarded as the most critical. The development of the Internet and mobile computing technologies has given rise to emerging applications such as teleconference, pay-TV, collaborating tasks, and message services [7, 9, 10, 12, 25, 29, 30]. Before obtaining the access to these services, resource providers have to delegate authority to legal subscribers. Hereafter, users can surf over the Internet or get online for communications via wired, wireless, 3G, or LTE networks. Traditional peer-to-peer communications do not suffice for these applications any more. Along with this trend, how to design high-performance peer-to-group and group-to-group communications has become an important research issue in heterogeneous networks [1, 4, 8, 11, 13, 15, 16, 17, 20, 22, 24, 26].

There are two main issues in designing a multicast mechanism: scalability and mobility. Scalability involves how to maintain the high performance in a large network for emerging applications, while mobility addresses how to efficiently complete key updates caused by frequent entrances and exits of members. As the fast development of networks has deeply affected the current world, more individuals are involved in this area with high frequency of mobility. Therefore, efficient solving plans of these two concerns are sure to contribute much to the network communications. Owing to the past researches, three main solutions have been proposed for providing secure multicast communications and key distribution: central control, distributed control, and subgroup control.

Central control: A central manager takes responsibility for the security of the entire group and key distribution. However, this solution is unsuitable for large networks, since the efficiency of the central manager will become the performance bottleneck of group communications. The failure of the central manager may lead to the inactive communication of the whole group [18, 31].

Distributed control: All group members take obligation for key generation and the security of the group. Since this solution is based on the Diffie-Hellman protocol, each group member must sustain exponential modulations for key updates caused by frequent

members joining and leaving [5, 6, 14, 19] , which makes it infeasible for large networks.

Subgroup control: The group is divided into several subgroups, each of which is controlled by a subgroup manager. The scalability of this approach is better than the other solutions, because the failure of one subgroup manger does not result in inactive overall group communication [2, 21, 28].

Although subgroup control is more feasible for providing multicast communications in large and high-mobility networks, there are still several weaknesses within this approach. First, each group user must keep many secret keys, which is quite inconvenient for involved participants considering the key storage ability. For example, if the mobile users want to launch a teleconference referring to this multicast mechanism, then more keys are needed to keep, much harder it is for them to communicate efficiently. Second, when a member joins or leaves the group, both involved and non-involved participants must change their secret keys to confirm the forward and backward secrecy. Taking the growing scale of the network into account, under this system one node action of joining or leaving happens, all the nodes have to handle heavy computation burthen. It is clear that this is so inefficient and unsuitable for large scale networks with countless participants currently. What is more, as in this system each time all the nodes have to take some actions, the energy wasted is worth thinking about. Thus, on the purpose of improving the efficiency of the whole system and achieving the energy saving, how to lower down the number of nodes taking actions for every time is as worth as a key factor. That is to say, reducing the number of participants for these cases becomes a critical challenge in designing a feasible multicast framework.

In particular, the computation overhead of frequent key updates also becomes a heavy burden for group members in a high-mobility environment. Seeing to the quick growth of the work burden and high development of the life quality, it is more and more popular to move fast to deal with emergent issues without extra time delay. Therefore, this noble mobility character requires the fast and precious disposal of the key updating and session key reconstruction. At the same time, it is sure that the security of the key and message need to be guaranteed firmly without falling down corresponding to the reducing of dealing time.

Out of all the considerations mentioned above, this article proposes a novel subgroup-based multicast protocol (SMP) adopting Lagrange Interpolating Polynomial (LIP) technique, which not only preserves the functionality of subgroup control mechanisms but also mitigates the encumbrance of group high-mobility. Furthermore, each group user only needs to keep one secret key in his/her database, compared with the traditional ones, in which the individual key, key encryption key and group key have to be stored as least. In addition, the number of participants involved in member joining and leaving can be effectively reduced to decrease the whole communication time and save power, making it portable for large scale networks. Moreover, owing to the highly efficient algorithms for updating and reconstructing keys when there joins a new node or leaves an existent node, the time consumption during these processes is considerable cut down. In this way, high-mobility network environment can refer to this efficient mechanism to attain flexible usages. Besides, simulation results will demonstrate that SMP outperforms other related mechanisms in scalable as well as high-mobility network environments.

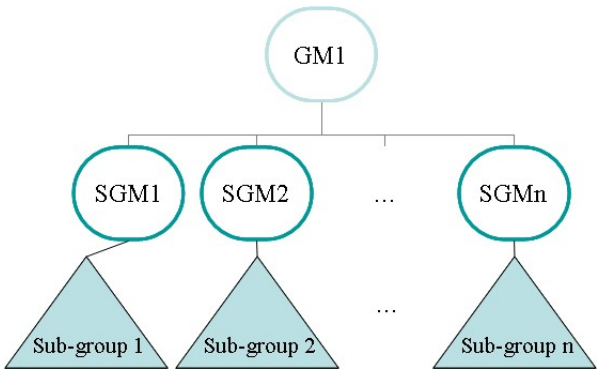


Figure 1: The structure of subgroup-based multicast

The rest of this paper is organized as follows. A model of subgroup-based multicast is described in Section 2, followed by the description of SMP in Section 3. Analyses of SMP are given in Section 4. Discussions and comparisons between other related works and SMP are shown in Section 5. Finally, we make conclusions in Section 6.

2 Model of Subgroup-based Multicast

The general structure of subgroup-based multicast protocols is illustrated in Figure 1. The main idea of these protocols is to divide the whole group into several subgroups. Each subgroup i is formed with a hierarchy structure and is controlled by a subgroup manager SGM_i, where $i = 1, 2, \cdots, n$ and n is the number of subgroups. The group manager GM shares a different secret key $K(GS_i)$ with each SGM_i and generates another secret key $K(GS)$ shared among all SGM_i's.

As shown in Figure 2, each internal node of subgroup i is a virtual node with a unique secret key and each leaf node denotes a subgroup member. Each member owns a private key and has to learn secret keys of the internal node on the path from the subgroup manager to himself/herself. For example, in Subgroup i, the user U_1 must know $K_i(h, 1)$, $K_i(h-1, 1)$, $\cdots$, $K_i(0, 1)$, where h is the height of subgroup i, and d is the maximum degree of each internal nodes.

Furthermore, several assumptions are made in the multicast system. First, when a new member wants to join

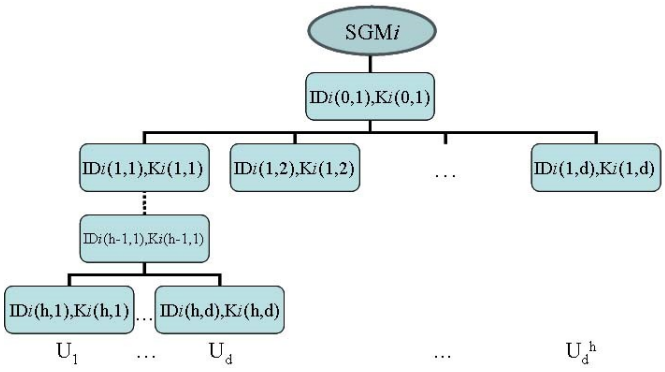


Figure 2: The hierarchy structure of Subgroup i

the group, GM must take responsibility for finding an empty place and generating a secret key for him/her. If all subgroups are full, GM has to create a new subgroup. Second, all nodes belonging to SGM_i are assumed to be trustworthy.

What is more, the subgroup managers ought to preserve the forward and backward secrecy of group communications to enhance the security of the system.

The Forward Secrecy: If a new member is permitted to join a subgroup, secret keys of the internal node on the path from the subgroup manager to himself/herself must be changed to prevent previous group messages from being learned by the new user.

The Backward Secrecy: In case that a member is expelled from the group, the subgroup manager has to modify secret keys of the internal node on the path from the node to its subgroup manager to stop the expellee from learning incoming group messages.

3 The Scalable Multicast Protocol (SMP)

Lagrange Interpolating Polynomial and a bulletin board are adopted in SMP to reduce the number of participants involved in member joining and leaving operations. Note that only GM and the legal SGM_i's can modify and update the bulletin board. The whole group is divided into several subgroups formed with hierarchy structures of height h. As illustrated in Figure 2, every node in the hierarchy structure of subgroup i is assigned a unique identity $ID_i(b,j)$ and a secret key $K_i(b,j)$, where $b = 0,1,\cdots,h$, $j = 1,2,\cdots,d$, and d is the maximum degree of the internal node in the hierarchy tree.

Before describing the broadcasting procedure of SMP, we first introduce the definition of Lagrange Interpolating Polynomial and how GM constructs the bulletin broad in Subsections 3.1 and 3.2.

3.1 Definition of Lagrange Interpolating Polynomial (LIP)

Let (x_1,y_1), (x_2,y_2), $\cdots$, (x_t,y_t) be t points on two-dimensional plane [21], N be a 128-bit prime, and a_0, $a_1,\cdots,a_{t-1}$ be integers ranged within [1, N-1]. To attain a polynomial $f(x)$, where y=$f(x)$ passes through the t points, we refer to the Lagrange Polynomial to calculate

$$f(x) = \sum_{j=1}^{t} y_j \prod_{i=1,i\neq j}^{t} \left(\frac{x-x_i}{x_j-x_i}\right) mod\ N$$
$$= a_0 + a_1x + a_2x^2 + ... + a_{t-1}x^{t-1} mod\ N.$$

Note that $y = f(x)$.

3.2 The Bulletin Board Setup

Here, we describe how GM constructs the bulletin board as shown in Table 1. For each internal node $ID_i(b,j)$ in subgroup i, SGM_i bottom-up computes a corresponding polynomial $ID_i(b,j)$_P(x) as follows, where b = h-1, h-2, $\cdots$, 0 and $j = 1,2,\cdots,d$.

Step 1: Computes d distinct hash values

$$\begin{aligned}
h_{ib_1} &= h(K_i(b+1,1), ID_i(b,j), ID_i(b+1,2), \\
&\quad \cdots, ID_i(b+1,d)), \\
h_{ib_2} &= h(K_i(b+1,2), ID_i(b+1,1), ID_i(b,j), \\
&\quad \cdots, ID_i(b+1,d)), \\
&\vdots \qquad\qquad \vdots \\
h_{ib_d} &= h(K_i(b+1,d), ID_i(b+1,1), \\
&\quad ID_i(b+1,2), ID_i(b+1,3), \cdots, \\
&\quad ID_i(b+1,d-1), ID_i(b,j)).
\end{aligned}$$

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates $(h_{ib_1}, K_i(b,j)+ h_{ib_1})$, $(h_{ib_2}, K_i(b,j) + h_{ib_2}, \cdots$, and $(h_{ib_d}, K_i(b,j) + h_{ib_d})$, to obtain the polynomial

$$ID_i(b,j)\_P(x) = a_0 + a_1x + a_2x^2 + ... + a_{d-1}x^{d-1} \bmod N,$$

where $a_0, a_1, \cdots, a_{d-1}$ are integers and $h(\cdot)$ is the secure one-way hash function.

Step 3: Publishes all identities of nodes and their corresponding polynomials on the bulletin board.

3.3 Message Broadcast Operation

While a message M needs to be broadcasted, GM randomly generates a new secret key $K(msg)$ to encrypt M. Next, GM computes the followings, $E_{K(msg)}[M]$ and $E_{K(GS)}[K(msg)]$, where $E_K[\cdot]$ is the AES-based symmetric encryption with secret key K. Then GM broadcasts the computation results to all SGM_i's. After receiving the messages, each SGM_i computes $K(msg)$ =

Table 1: The example of the bulletin board

SGM_i

Nodes	Polynomiials
$ID_i(b,1)$	$ID_i(b,1)\_P(x)$
$ID_i(b,2)$	$ID_i(b,2)\_P(x)$
⋮	⋮
$ID_i(b,d)$	$ID_i(b,d)\_P(x)$

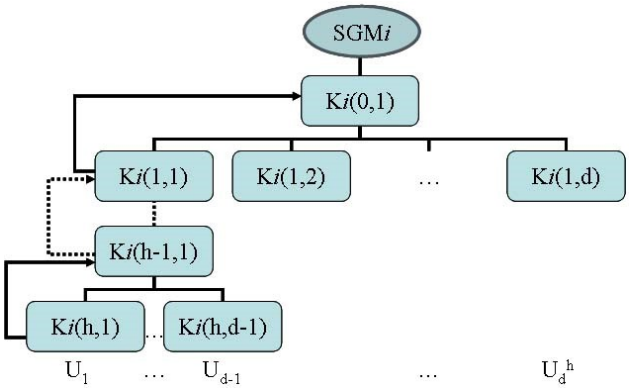


Figure 3: The example of message broadcast operation in subgroup i

$D_{K(GS)}[E_{K(GS)}[K(msg)]]$ and $E_{K_{i(0,1)}}[K(msg)]$, $D_K[\cdot]$ is the AES-based symmetric decryption with secret key K and $K_{i(0,1)}$ is the common secret key shared by all members in subgroup i. SGM_i then broadcasts the following messages to all subgroup users, $E_{K_{i(0,1)}}[K(msg)]$ and $E_{K(msg)}[M]$.

As shown in Figure 3, the subgroup member U_1 uses his/her secret key to obtain the secret key of the upper level by computing $K_i(h-1,1) = ID_i(h-1,1)\_P(h_{ib_1}) - h_{ib_1}$, where $h_{ib_1} = h(K_i(h,1),\ ID_i(h-1,1),\ ID_i(h,2),\ ID_i(h,3),\ \cdots,\ ID_i(h,d-1))$ is pre-computed by U_1 and b = h-1. By the same way, U_1 can quickly obtain $K_i(0,1)$ to retrieve $K(msg)$ and decrypt the message M.

4 Analyses of SMP

The previous section describes the normal operation of broadcasting a group message in static. The proliferation of the Internet and mobile computing technologies, however, makes the membership of a group vary from minute to minute. Therefore, we analyze how SMP manipulates the changes of membership and network topologies, which lead to much more frequent key update. Here in SMP, we mainly consider three types of mobility: subgroup manager joining, member joining, and member leaving.

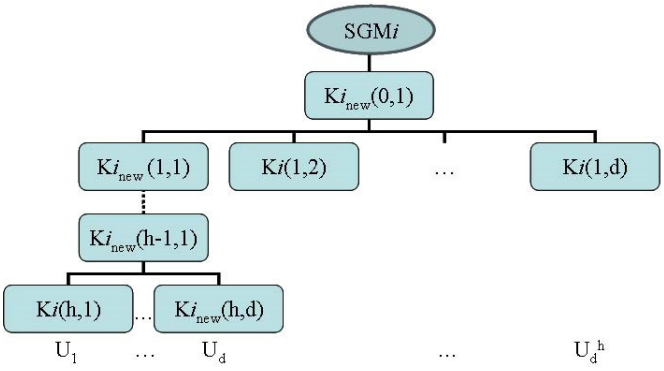


Figure 4: The example of member join operation in SMP

4.1 Subgroup Manager Joining Operation

If the scale of system users exceeds in the size of whole group, GM has to designate a new subgroup manager SGM_{n+1} and change the secret key from $K(GS)$ to $K_{new}(GS)$. Besides, GM has to generate a new secret key $K(GS_{n+1})$ shared between GM and SGM_{n+1}. Next, GM computes $E_{K(GS)}[K_{\text{new}}(GS)]$ and $E_{K(GS_{n+1})}[K_{new}(GS)]$.

GM then broadcasts the computation results to all SGM_i's including the new one. While receiving the messages, the original SGM_i's retrieve the new secret key $K_{new}(GS)$ by computing $D_{K(GS)}[E_{K(GS)}[K_{new}(GS)]]$.

On the other hand, the new subgroup manager retrieves the new secret key $K_{new}(GS)$ by calculating $D_{K(GS_{n+1})}[E_{K(GS_{n+1})}[K_{new}(GS)]]$.

Hence, the joining operation of a new subgroup manager is completed.

4.2 Member Joining Operation

While a new member U_d wants to join the communication group, GM has to find a suitable place and generate a secret key $K_{i_{new}}(h,d)$ for him/her. As illustrated in Figure 4, all secret keys of the path from SGM_i to U_d must be modified to preserve the forward secrecy. The secret key $K_i(b,1)$ must be changed, where $b = 0,1,\cdots,h-1$. All involved internal nodes' polynomials published on the bulletin board will be updated by SGM_i. That is, SGM_i has to bottom-up perform Lagrange Interpolating Polynomial $(h-1)$ times to reconstruct $(h-1)$ involved polynomials.

For each involved internal node $ID_i(b,1)$, where $b = h-1,\ h-2,\ \cdots,\ 0$ (i.e. the internal nodes on the path from U_d to SGM_i), SGM_i executes the followings.

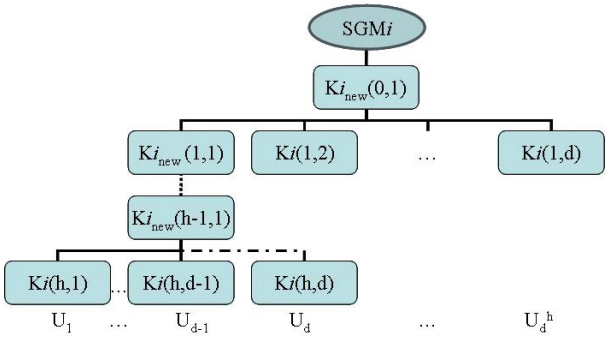


Figure 5: The example of member leave operation in SMP

Step 1: Computes d distinct hash values

$$\begin{aligned}
h_{ib_1} &= h(K_i(b+1,1), ID_i(b,1), ID_i(b+1,2), \\
&\quad \cdots, ID_i(b+1,d)), \\
h_{ib_2} &= h(K_i(b+1,2), ID_i(b+1,1), ID_i(b,1), \\
&\quad ID_i(b+1,3), ID_i(b+1,4), \\
&\quad \cdots, ID_i(b+1,d)), \\
&\vdots \qquad\qquad \vdots \\
h_{ib_{(d-1)}} &= h(K_i(b+1,d-1), ID_i(b+1,1), \\
&\quad ID_i(b+1,2), ID_i(b+1,3), \cdots, \\
&\quad ID_i(b+1,d-2), ID_i(b,1)), \\
&\quad ID_i(b-1,d)), \\
h_{ib_d} &= h(K_i(b+1,d), ID_i(b+1,1), \\
&\quad ID_i(b+1,2), ID_i(b+1,3), \cdots, \\
&\quad ID_i(b+1,d-1), ID_i(b,1)).
\end{aligned}$$

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates $(h_{ib_1}, K_i(b,1) + h_{ib_1})$, $(h_{ib_2}, K_i(b,1) + h_{ib_2}, \cdots,$ and $(h_{ib_d}, K_i(b,1) + h_{ib_d})$, to obtain the polynomial

$$ID_i(b,1)\_P(x) = a'_0 + a'_1 x + \cdots + a'_{d-1} x^{d-1} \bmod N,$$

where $a'_0, a'_1, \cdots, a'_{d-1}$ are integers.

Step 3: Updates the modified information on the bulletin board as shown in Table 1.

4.3 Member Leaving Operation

While a user U_d leaves Subgroup i, as illustrated in Figure 5, all secret keys of the path from SGM_i to U_d must be modified to preserve the backward secrecy. The secret key $K_i(b,1)$ must be changed, where $b = 0, 1, \cdots, h-1$. All involved internal nodes' polynomials published on the bulletin board have to be modified by SGM_i in time. That is, SGM_i needs to bottom-up perform Lagrange Interpolating Polynomial $(h-1)$ times to reconstruct $(h-1)$ involved polynomials.

For each involved internal node $ID_i(b,1)$, where $b = h-1, h-2, \cdots, 0$ (i.e. the internal nodes on the path from U_d to SGM_i), SGM_i executes the followings.

Step 1: Computes

$$\begin{aligned}
h_{ib_1} &= h(K_i(b+1,1), ID_i(b,1), ID_i(b+1,2), \\
&\quad \cdots, ID_i(b+1,d-1)) \\
h_{ib_2} &= h(K_i(b+1,2), ID_i(b+1,1), ID_i(b,1), \\
&\quad ID_i(b+1,3), ID_i(b+1,4), \cdots, \\
&\quad ID_i(b+1,d-1)) \\
&\vdots \qquad\qquad \vdots \\
h_{ib_{d-1}} &= h(K_i(b+1,d-1), ID_i(b+1,1), \\
&\quad ID_i(b+1,2), ID_i(b+1,3), \cdots, \\
&\quad ID_i(b+1,d-2), ID_i(b,1)).
\end{aligned}$$

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates $(h_{ib_1}, K_{i_{new}}(b,1) + h_{ib_1})$, $(h_{ib_2}, K_{i_{new}}(b,1) + h_{ib_2}, \cdots,$ and $(h_{ib_{(d-1)}}, K_{i_{new}}(b,1) + h_{ib_{(d-1)}})$, to obtain the polynomial

$$\begin{aligned}
ID_i(b,1)_P(x) &= a''_0 + a''_1 x + a''_2 x^2 + \cdots \\
&\quad + a''_{d-1} x^{d-1} \bmod N,
\end{aligned}$$

where $a''_0, a''_1, \cdots, a''_{d-1}$ are integers.

Step 3: Updates the modified information on the bulletin board.

5 Discussions

We discuss how SMP is able to preserve the forward and backward secrecy during group communications and confirm the broadcast of messages at length in this section. Furthermore, we also present the performance comparisons with related works of SMP to demonstrate its convenience, low computation cost, and low storage space.

5.1 Security Examination

Since the member joining operation and member leaving operation are always performed frequently in dynamic network groups, it is important to prevent messages from being illegally shared. The security of SMP is based on three cryptographic assumptions.

Secure AES-based symmetric en/decryption $[m]_k$.
With the message m, it is relatively easy to encrypt m as $[m]_k$ with an AES-based symmetric key k, while it is computationally infeasible to retrieve m from $[m]_k$ without the knowledge of k.

Discrete logarithm assumption.
Given a generator g, a large prime p, and a random number $x \in Z_p$, it is easy to compute $y = g^x \bmod p$. Then it is computationally infeasible to compute x just referring to y, g, and p. Note that x is called the discrete logarithm of y with respect to g.

Secure one-way hash function $h(\cdot)$.

1) Preimage resistance: As to a message m, it is easy to compute $h(m)$; nevertheless, it is computationally infeasible to attain m only from the knowledge of $h(m)$.
2) 2nd-preimage resistance: According to $h(m)$, it is impossible to find m' such that $h(m) = h(m')$ except applying the brute-force method.

5.1.1 Preserving the Forward Secrecy

As depicted in Figure 4, during the entrance of a new member U_d, he/she is assigned an identity $ID_{i_{new}}(h,d)$ and a secret key $K_{i_{new}}(h,d)$, which U_d can make use of to construct secret keys, $K_{i_{new}}(h-1,1)$, $K_{i_{new}}(h-2,1)$, $\cdots$, $K_{i_{new}}(0,1)$. To keep U_d from learning previous group messages, SMP must prevent U_d from figuring out the secret keys, $K_i(h-1,1)$, $K_i(h-2,1)$, $\cdots$, $K_i(0,1)$, shown in Figure 3. Here, we prove that SMP can confirm the forward secrecy by Proposition 1.

Proposition 1. *If a new member U_d wants to learn the previous messages shared between old group members, he/she must fail.*

Proof. To learn the advanced messages, U_d must first obtain one of the hash values h_{ib_1}, h_{ib_2}, $\cdots$, $h_{ib_{d-1}}$, where $b = h-1$. Then, U_d can apply the hash value to $ID_i(h-1,1)\_P(x)$ in order to retrieve $K_i(h-1,1)$. Given $K_{i_{new}}(h,d)$, $ID_i(h-1,1)$, $ID_i(h,2)$, $ID_i(h,3)$, $\cdots$, $ID_i(h,d-1)$, $ID_{i_{new}}(h,d)$, U_d can obtain the hash value,

$$\begin{aligned} h^*_{ib_d} &= h(K_{i_{new}}(h,d), ID_i(h,1), ID_i(h,2), ID_i(h,3), \\ &\quad \cdots, ID_i(h,d-1), ID_i(h-1,1)). \end{aligned}$$

Since

$$\begin{aligned} h_{ib_1} &= h(K_i(h,1), ID_i(h-1,1), ID_i(h,2), \\ &\quad ID_i(h,3), \cdots, ID_i(h,d-1)), \\ h_{ib_2} &= h(K_i(h,2), ID_i(h,1), ID_i(h-1,1), \\ &\quad ID_i(h,3), \cdots, ID_i(h,d-1)), \\ &\vdots \qquad\qquad \vdots \\ h_{ib_{d-1}} &= h(K_i(h,d-1), ID_i(h,1), ID_i(h,2), \cdots, \\ &\quad ID_i(h,d-2), ID_i(h-1,1)). \end{aligned}$$

We infer that $h^*_{ib_d}$ must be different from h_{ib_1}, h_{ib_2}, $\cdots$, and $h_{ib_{(d-1)}}$ under the assumption of the secure one-way hash function. That is, U_d cannot apply $K_{i_{new}}(h,d)$ and $ID_i(h-1,1)\_P(x)$ to obtain $K_i(h-1,1)$. Similarly, U_d is unable to learn $K_i(h-2,1)$, $K_i(h-3,1)$, $\cdots$, and $K_i(0,1)$.

Again, since N is a large prime, if U_d wants to resolve $ID_i(h-1,1)\_P(x)$ without one of the hash values h_{ib_1}, h_{ib_2}, $\cdots$, and $h_{ib_{d-1}}$, he/she must face the difficulty of solving the discrete logarithm problem. It is computationally infeasible for U_d to achieve this attempt. Furthermore, since we assume a secure AES-based symmetric en/decryption in the multicast system, U_d cannot compromise the previous group messages without $K_i(0,1)$. □

5.1.2 Preserving the Backward Secrecy

As illustrated in Figure 5, when U_d is expelled from subgroup i, SMP has to prevent U_d from listening to group communications continually. Consequently, SGM_i must modify the secret keys of the internal node on the path from SGM_i to U_d. That is, SGM_i must reconstruct secret keys $K_{i_{new}}(h-1,1)$, $K_{i_{new}}(h-2,1)$, $\cdots$, and $K_{i_{new}}(0,1)$ and then apply them to update $ID_i(h-1,1)\_P(x)$ on the public board. We demonstrate that SMP can confirm the backward secrecy by Proposition 2.

Proposition 2. *If an expellee U_d wants to learn the content of the incoming messages shared among current group members, he or she must fail.*

Proof. To uncover incoming group messages, U_d must firstly obtain one of the hash values

$$h^*_{ib_1}, h^*_{ib_2}, \cdots, h^*_{ib_{d-1}},$$

where $b = h-1$,

$$\begin{aligned} h^*_{ib_1} &= h(K_i(h,1), ID_i(h-1,1), ID_i(h,2), \\ &\quad ID_i(h,3), \cdots, ID_i(h,d-1)), \\ h^*_{ib_2} &= h(K_i(h,2), ID_i(h,1), ID_i(h-1,1), \\ &\quad ID_i(h,3), \cdots, ID_i(h,d-1)), \\ &\vdots \qquad\qquad \vdots \\ h^*_{ib_{d-1}} &= h(K_i(h,d-1), ID_i(h,1), ID_i(h,2), \cdots, \\ &\quad ID_i(h,d-2), ID_i(h-1,1)). \end{aligned}$$

It is clear that SGM_i does not use h_{ib_d} and $K_i(h,d)$ to update $ID_i(h-1,1)\_P(x)$. Hence, it is computationally infeasible for U_d to solve $ID_i(h-1,1)\_P(x)$ without the correct hash values, under the assumption of the discrete logarithm. Furthermore, it is also computationally infeasible to compute a hash value that equals to $h^*_{ib_1}$, $h^*_{ib_2}$, $\cdots$, or $h^*_{ib_{d-1}}$ under the assumption of the secure one-way hash function. Since we assume a secure AES-based symmetric en/decryption in multicast systems, U_d cannot listen to incoming group messages without knowing $K_{i_{new}}(0,1)$. □

5.2 Performance Evaluations and Comparisons

In the following subsection, we compare SMP with related works to show its outstanding advantages. Notations used in Table 2 are defined as follows.

n: total number of subgroups;

m: total number of subgroup members;

d: maximum degree of each internal node in the hierarchy structure;

h: height of each subgroup ($m = d^h$);

$h_1 : log_d(n \times m)$;

Table 2: Comparisons with related works

		[12]	[2]	[31]	[21]	SMP
Member join	SGM	$2(h{+}1)$K	$2h$K	$2h_1$K	2K	$(h$-1)P+(hd)H
	IN	$(h{+}1)$K	hK	h_1K	1K	0
	NON	$(d/d-1)$K	$(d/d-1)$K	$(d/d-1)$K	1K	0
Member leave	SGM	$[2(hd)+(h-d)]$K	$[2(hd-1)+(h-d-1)]$K	$[2(h_1d-1)+(h_1-d-1)]$K	$(m-1)$K	$(h$-1)P+(hd)H
	IN	-	-	-	-	-
	NON	$(3d/d-1)$K	$(3d/d-1)$K	$(3d/d-1)$K	1K	0
Message broadcast	SGM	2K	2K	2K	2K	2K
	IN	1K	1K	1K	1K	hY+1K
	NON	-	-	-	-	-

IN: involved member;

NON: non-involved member;

K: computation overhead of a symmetric en/decryption operation;

H: computation overhead of a one-way hash operation;

P: computation overhead of constructing the polynomial by LIP;

Y: computation overhead of obtaining y with input x, where $y = f(x)$;

To access broadcast messages and update subgroup session keys, both involved and non-involved members must perform cryptographic operations frequently within the system. Excluding the security consideration, whether it is able to achieve these operations efficiently often dominates the evaluation of a multicast mechanism. As illustrated in Table 2, SMP adopts the Lagrange Interpolating Polynomial and the secure one-way hash function in addition to the symmetric en/decryption algorithm.

As mentioned in [28], a one-way hash function can be executed at least 10 times faster than a symmetric en/decryption. It is clear that the computation overhead of constructing the polynomial by LIP is quite lighter than that of performing the symmetric en/decryption, since the construction of LIP polynomial is based on simple multiplication, while the en/decryption function needs round operations [3, 6, 27]. Hence, SMP outperforms other works in terms of member joining and member leaving operations.

Specifically, when a member joins or leaves the subgroup, all subgroup members must update session keys to confirm the forward and backward secrecy as presented in [2, 12, 21, 31]. In SMP, by contrast, only the SGM has to construct and update corresponding polynomials on the bulletin board. This can effectively lower down the computation overhead of group members. To preserve this benefit, involved users must perform $h^*Y + 1K$ operations to retrieve a broadcast message; users in other works only need to execute $1K$ operation. The simulation results can show that this extra overhead is still acceptable.

5.3 Simulation Results

As is shown, simulation results demonstrate the superiority of SMP over its predecessors. Simulators were executed in the VC6.0 language. Cryptographic routines, including the AES algorithm and SHA-1, were implemented using the public OpenSSL library [23]. Note that the processing time of the simulation does not include the time used to authenticate the new member. That is, only the computation overhead of obtaining broadcasted message and constructing session keys is taken into consideration in the simulation. The symmetric en/decryption algorithm and the secure one-way hash function used in simulators are AES-128 and SHA-1, respectively. The encrypted message is 512k bytes, and the large prime N adopted in LIP is 128 bits. The ratio of member joining and member leaving to message broadcasting operation is set to 1:1:1.

To begin, we set the total number of subgroups (n to four) and the degree of each internal node in the hierarchy structure (d to two). The average SGM processing time (AvrST) is the mean of the total time needed for SGM to complete one member joining operation, one member leaving operation, and one message broadcasting operation. The average IN processing time (AvrIT) is the mean of the time consumption that takes involved nodes to complete one member joining operation and obtain one broadcasted message. The average NON processing time (AvrNT) is the mean of the time charge that takes a non-involved member to complete one member joining operation and one member leaving operation.

Table 3 compares SMP with related works in terms

Table 3: AvrST/AvrIT/AvrNT vs. m (512K bytes)

d=2, 512k		m(nodes)						
		64	128	256	512	1024	2048	4096
AvrST (seconds)	[12]	0.3128	0.3477	0.3826	0.4175	0.4524	0.4873	0.5222
	[2]	0.1947	0.2295	0.2648	0.2991	0.3341	0.3683	0.4037
	[31]	0.2643	0.2992	0.3339	0.3687	0.4032	0.4381	0.4729
	[21]	0.3338	0.6517	1.2872	2.5582	5.1002	10.1851	20.3541
	SMP	0.0181	0.0198	0.0223	0.0241	0.0254	0.0271	0.0287
AvrIT (seconds)	[12]	0.0538	0.0581	0.0622	0.0664	0.0705	0.0747	0.0789
	[2]	0.0352	0.0398	0.0451	0.0499	0.0559	0.0598	0.0657
	[31]	0.0459	0.0501	0.0559	0.0601	0.0657	0.0699	0.0758
	[21]	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101
	SMP	0.0062	0.0062	0.0063	0.0063	0.0064	0.0065	0.0065
AvrNT (seconds)	[12]	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099
	[2]	0.0397	0.0397	0.0397	0.0397	0.0397	0.0397	0.0397
	[31]	0.0397	0.0397	0.0397	0.0397	0.0397	0.0397	0.0397
	[21]	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099
	SMP	0	0	0	0	0	0	0

Table 4: AvrST/AvrIT/AvrNT vs. m (1M bytes)

d=2, 1M		m(nodes)						
		64	128	256	512	1024	2048	4096
AvrST (seconds)	[12]	0.6631	0.7316	0.8001	0.8686	0.9371	1.0056	1.0741
	[2]	0.3849	0.4537	0.5227	0.5914	0.6599	0.72791	0.7979
	[31]	0.5224	0.5914	0.6599	0.7287	0.7978	0.8665	0.9357
	[21]	0.6598	1.2897	2.5484	5.0668	10.1028	20.1754	40.3198
	SMP	0.0288	0.0299	0.0321	0.0337	0.0353	0.0367	0.0384
AvrIT (seconds)	[12]	0.0975	0.1076	0.1177	0.1278	0.1379	0.148	0.1581
	[2]	0.0697	0.0798	0.0895	0.0997	0.1094	0.1192	0.1289
	[31]	0.895	0.0997	0.1094	01192	0.1291	0.1388	0.1487
	[21]	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207
	SMP	0.0114	0.0114	0.0114	0.0114	0.0114	0.0114	0.0114
AvrNT (seconds)	[12]	0.0199	0.0199	0.0199	0.0199	0.0199	0.0199	0.0199
	[2]	0.0791	0.0791	0.0791	0.0791	0.0791	0.0791	0.0791
	[31]	0.0791	0.0791	0.0791	0.0791	0.0791	0.0791	0.0791
	[21]	0.0199	0.0199	0.0199	0.0199	0.0199	0.0199	0.0199
	SMP	0	0	0	0	0	0	0

of AvrST/AvrIT/AvrNT versus the number of nodes m. Furthermore, the input size of AES-128 in Table 3 is set to 512k bytes, while that in Table 4 is set to 1M bytes. Owing to this simulating method, it can be concluded that the security of the whole system is able to be enhanced without prolonging the time consumption for member joining/leaving and message broadcasting process. As is shown in Table 3, the time needed by SMP for AvrST is only 0.0181 seconds, while the lowest time cost of related works is still 0.1947 seconds under the condition that there are 64 nodes. Even though the time charge as AvrST grows in all theses methods along with the increase of nodes, SMP just needs 0.0287 seconds for 4096 nodes, while that is 0.4037 seconds at least in the related works. It is clear that SMP outperforms other approaches in all listed cases.

Although involved users in SMP must perform extra h*Y operations in order to retrieve a broadcast message, simulation results prove that the extra overhead is acceptable and still leave SMP the most optimal method among all the related multicast mechanisms. As is seen, it is the same with the AvrIT. In addition, all AvrNT values are zero in SMP, which confirms that in this method non-involved nodes do not need extra processing time in member joining/leaving operation. This advantage can effectively reduce the bandwidth consumption and computation overheads of non-involved nodes while maintaining high network mobility. As mentioned above, while in Table 4 the inputted size of AES-128 is set to 1M bytes, we can obtain the similar results. That is, no matter how the number of nodes rises, all the costs of SMP are the least in AvrST, AvrIT and AvrNT among related works. Since time depletion is the key factor for multicasting systems, this obvious advantage of time saving for SMP makes it more efficient in dynamic networks.

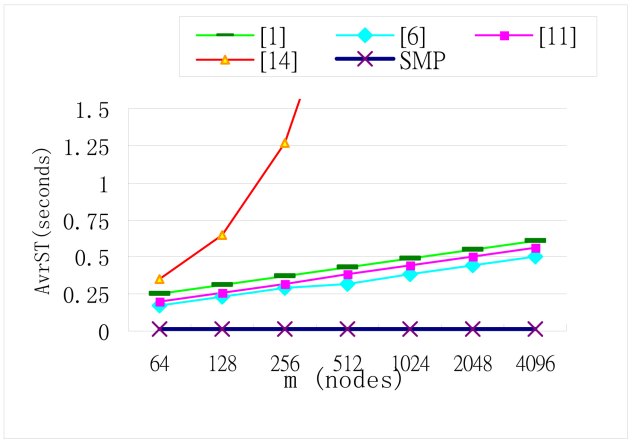
Besides, we adjust $d = 4$ to demonstrate the practicability of SMP in Figure 6. The inputted size of AES-128 in Figures 6(a), (b), and (c) is 512k bytes, while that in Figures 6(d), (e), and (f) is 1M bytes to show the security guaranteeing of SMP. As is displayed, Figure 6 shows the comparisons of SMP with related works in terms of AvrST/AvrIT/AvrNT versus m. From the trends shown in Figure 6, it is obvious that SMP still outperforms the works of [12, 2, 31, 21] considering the computation cost and communication cost, which are based on the time consumption. In Figure 6(a), the AvrST of SMP ranges from 0.0143 seconds to 0.0239 seconds; while in Figure 6(b), the AvrIT of SMP ranges from 0.0064 seconds to 0.0067 seconds.

Compared with the time consumption of related works [2, 12, 21, 31] , in which the lowest AvrST is 0.315 (second) and the lowest AvrIT is 0.0214 (second), it implies that SMP has more advantages under this condition. Moreover, as to the AvrNT, SMP costs nothing with m raging from 64 to 4096. That means, within SMP the non-involved nodes need not to perform any computation or related actions for updating session keys during the node joining/leaving process. In this way, the computation and communication are largely lowered down for the whole system. Furthermore, concerned by the insecurity within all kinds of networks, the encryption algorithm and the length of the key have to be cared more seriously. According to Figure 6(a) and Figure 6(d), along with the extension of the AES-128, the time consumption in SMP just grows 0.0213 seconds to 0.0317 seconds for AvrST, which rises a little. While, for the related works, this time charge is magnified on an obvious level. The similar situations appear in AvrIT and AvrNT, as shown in Figure 6(b) vs. Figure 6(e) and Figure 6(c) vs. Figure 6(f), which emphasize that SMP can hold with the extension of secret key to enhance the reality of the whole multicasting system.

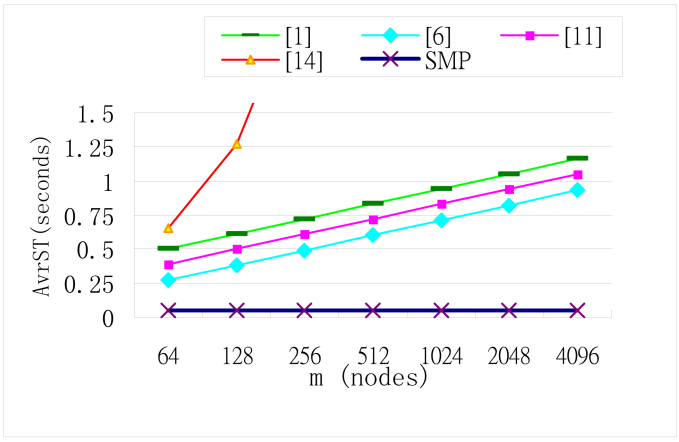
As is shown in the simulation, SMP can effectively reduce the overhead of SGM in broadcasting messages and updating session keys. It can also reduce the overhead of involved nodes in joining a subgroup and obtaining broadcast messages. Furthermore, SMP can decrease the overhead of non-involved nodes in completing member joining/leaving operations.

In addition, taking the key storage cost into account, each group participant in SMP only needs to keep one secret key in the database compared with the traditional methods, where at least the individual key, key encryption key and group key have to be stored. The pertinent comparisons of SMP and other works [2, 12, 21, 31] versus m are shown in Figure 7. Here we set the degree of each internal node in the hierarchy structure, d=4. It can be seen that participants in SMP are able to store only their secret keys to complete member joining, member leaving as well as message broadcasting operations, while those in others still have to store more extra keys. Here we evaluate the key storage cost of SMP and related works basing on [12]. As [12] proposes an optimal key tree structure to reduce the storage burden in traditional methods, we set it as the base to display the times that each node in other related works and SMP has of it, called the Times of Key Storage (ToKS).

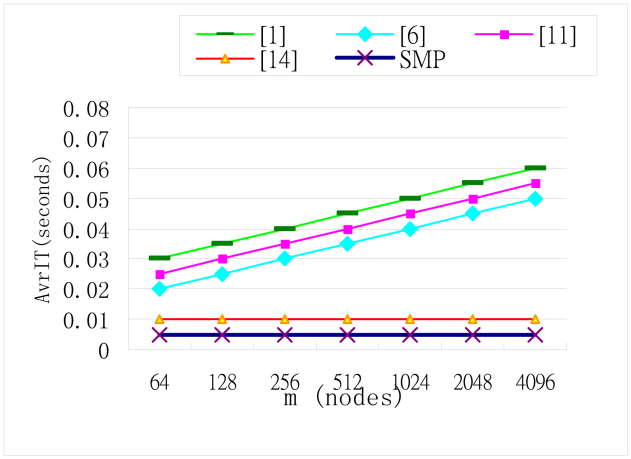
Figure 7 shows that related words all have several times of [12] in ToKS. And together with the increasing number of nodes from 64 to 4096, the key storage burden of related works grows almost 3 times. On the contrary, thanks to the LIP and subgroup mechanism, SMP achieves a much lower rate of [12] when nodes ranging from 64 to 4096. What is more, as in SMP each participant has to keep only one secret key, the key storage of users stays static no matter how many nodes there are. Therefore, as is displayed in Figure 7, SMP is able to save more space on the key storage management, which character is not affected by the number of participants. As is known to all, it can never be ignored to simultaneously reduce the overhead of communication cost as well as that of the key storage cost. Therefore, by successfully reducing the key storage cost, SMP outperforms other related mechanisms as the simulation results demonstrated.



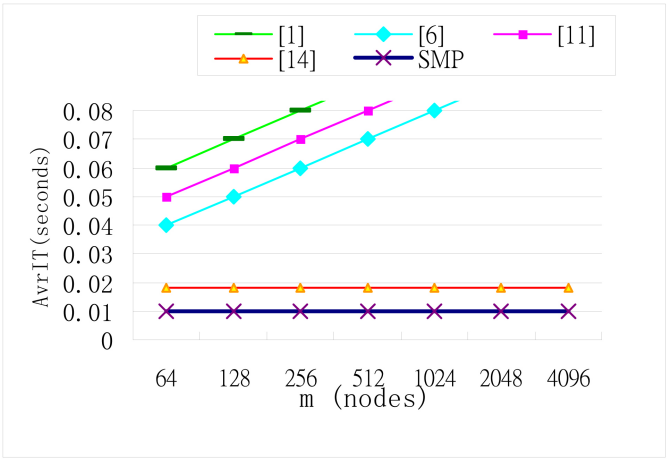
(a)



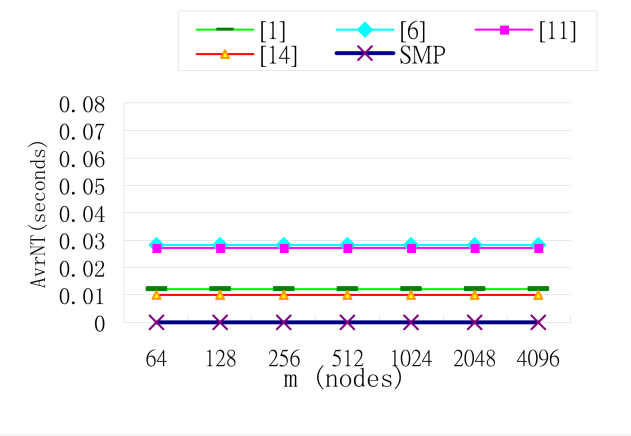
(d)



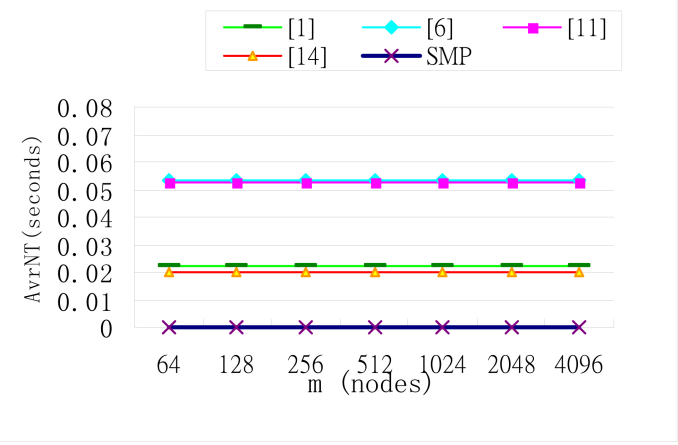
(b)



(e)



(c)



(f)

Figure 6: AvrST/AvrIT/AvrNT vs. $m, d = 4$

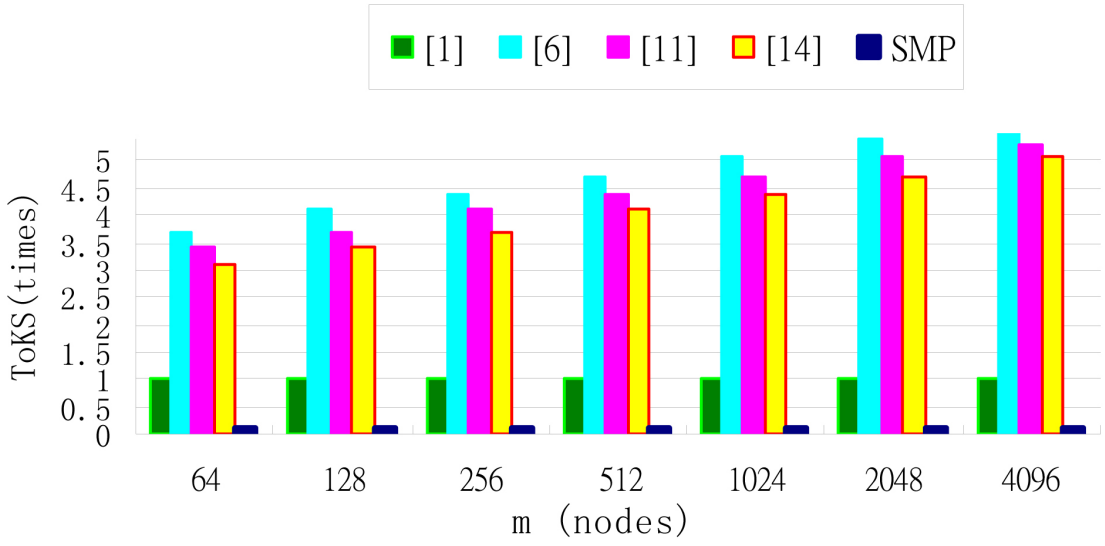


Figure 7: Key storage cost vs. $m, d = 4$

6 Conclusions

On all accounts, the subgroup control solution is more feasible for granting authority in large and high-mobility networks. However, we find that each subgroup member in the multicast system must keep many secret keys in his/her database. Furthermore, when a member joins or leaves the group, involved participants must modify their secret keys to preserve the forward and backward secrecy. In SMP, these disadvantages can be effectively improved by LIP technique. Specifically, simulation results have shown that SMP outperforms other works in terms of broadcasting messages and updating session keys to reduce communication cost, computation cost, and storage space. Therefore, SMP is more suitable for being applied to a large dynamic network environment.

References

[1] D. S. AbdElminaam, H. M. A. Kader, M. M. Hadhoud, and S. M. El-Sayedr, "Increase the performance of mobile smartphones using partition and migration of mobile applications to cloud computing," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 34–44, 2014.

[2] H. K. Aslan, "A scalable and distributed multicast security protocol using a subgroup-key hierarchy," *Computers and Security*, vol. 23, pp. 320–329, 2004.

[3] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[4] C. Campolo, C. Casetti, C.F. Chiasserini, and A. Molinaro, "A multirate MAC protocol for reliable multicast in multihop wireless networks," *Computer Networks*, vol. 56, no. 5, pp. 1554–1567, 2012.

[5] T. Y. Chang and M. S. Hwang, "User-anonymous and short-term conference key distribution system via link-layer routing in mobile communications," *International Journal of Mobile Communications*, vol. 9, no. 2, pp. 144–158, 2011.

[6] D. W. Davies, "Some regular properties of DES," in *Advances in Cryptology (Crypto'82)*, pp. 89–96, Springer, 1983.

[7] G. G. Deverajan and R. Saravanan, "A novel trust based system to detect the intrusive behavior in MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 31–43, 2015.

[8] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, 1976.

[9] S. M. El-Sayed, H. M. A. Kader, M. M. Hadhoud, and D. S. AbdElminaam, "Mobile cloud computing framework for elastic partitioned/modularized applications mobility," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 53–63, 2014.

[10] D. He, C. Chen, M. Ma, S. Chan, and J. Bu, "A secure and efficient password-authenticated group key exchange protocol for mobile ad hoc networks," *International Journal of Communication Systems*, vol. 26, no.4, pp. 495–504, 2013.

[11] I. Ingemarsson, D. Tang, and C. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 714–720, 1982.

[12] D. H. Je and S. W. Seo, "New key tree management protocol for the efficiency of storage and computation time in secure multicast communication," in *Proceedings of Wireless VITAE*, pp. 707–711, Aalbor, Denmark, 2009.

[13] G. Kadir, T. Kuseler, and I. A. Lami, "SMPR: a smartphone based MANET using prime numbers to enhance the network-nodes reachability and security of

routing protocols," *International Journal of Network Security*, vol. 18, no. 3, pp. 579–589, 2016.
[14] A. Kumar and S. Tripathi, "Anonymous ID-based Group Key Agreement Protocol without Pairing," *International Journal of Network Security*, vol. 18, no. 2, pp. 263–273, 2016.
[15] J. S. Lee, P. Y. Lin, and C. C. Chang, "Lightweight secure roaming mechanism between GPRS/UMTS and wireless LANs," *Wireless Personal Communications*, pp. 569–580, 2010
[16] G. Li, Q. Jiang, Y. Shi, and F. Wei, "Anonymous network information acquirement protocol for mobile users in heterogeneous wireless networks," *International Journal of Network Security*, vol. 18, no. 1, pp. 193–200, 2016.
[17] Y. Li and I. R. Chen, "Hierarchical agent-based secure and reliable multicast in wireless mesh networks, " *Computer Communications*, vol. 36, no. 14, pp. 1515–1526, 2013.
[18] Y. Li and I. R. Chen, "Dynamic agent-based hierarchical multicast for wireless mesh networks,," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1683–1698, 2013.
[19] Z. Li, C. Wang, C. Jiang, and X. Li, "Multicast capacity scaling for inhomogeneous mobile ad hoc networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 29–38, 2013.
[20] D. Manivannan and P. Neelamegam, "An efficient key management scheme in multi-tier and multi-cluster wireless sensor networks," *International Journal of Network Security*, vol. 17, no. 6, pp. 651–660, 2015.
[21] S. Mittra, "IOLUS: A framework for scalable secure multicasting," in *Proceedings of ACM SIGCOM*, pp. 277–278, Cannes, France, 1997.
[22] N. Omheni, F. Zarai, M. S. Obaidat, K.-F. Hsiao, and L. Kamoun, "A novel media independent handover-based approach for vertical handover over heterogeneous wireless networks," *International Journal of Communication Systems*, vol. 27, no. 5, pp. 811–824, 2014.
[23] The OpenSSL Project, 2016. (`http://www.openssl.org`)
[24] P. J. Pinero, J. A. Cortes, J. Malgosa, F. J. Canete, P. Manzanares, and L. Diez, "Analysis and improvement of multicast communications in HomePlug AV-based in-home networks," *Computer Networks*, vol. 62, pp. 89–100, 2014.
[25] A. Pinto and J. W. Atwood, "Secure multicast in IPTV services," *Computer Networks*, vol. 54, no. 10, pp. 1531–1542, 2010.
[26] Y. Rengasamy and A. Magrica, "Priority based resource allocation in hybrid network environment," *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 81–90, 2015.
[27] A. Shamir, "How to share a secret," *Communications of the ACM*, pp. 612–613, 1976.
[28] A. Shimizu and S. Miyaguchi, "Fast data encipherment algorithm FEAL," in *Advances in Cryptology (EUROCRYPT'87)*, pp. 267–278, Springer, 1987.
[29] C. Y. Sun and C. C. Chang, "Cryptanalysis of a secure and efficient authentication scheme for access control in mobile pay-TV systems," *International Journal of Network Security*, vol. 18, no. 3, pp. 594–596, 2016.
[30] D. Tian, J. Zhou, Y. Wang, H. Xia, Z. Yi, and H. Liu, "Optimal epidemic broadcasting for vehicular ad hoc networks," *International Journal of Communication Systems*, vol. 27, no. 9, pp. 1220–1242, 2014.
[31] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

Kuo-Jui Wei received the MS degree in information engineering and computer science in 2011. He is currently pursuing her Ph.D. degree in Information Engineering and Computer Science in Feng Chia University, Taichung, Taiwan. His current research interests include information security and mobile communications.

Jung-San Lee received his Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan in 2008. Since 2012, he has worked as an associate professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taichung, Taiwan. His current research interests include image processing, information security, and mobile communications.

Bo Li received the Bachelor degree of information security from Tongji University, China in 2011. She is currently pursuing her Ph.D. degree in Electrical Engineering and Computer Science in Vanderbilt University, Tennessee, USA. Her current research interests include secret sharing technique and mobile communications.