Dynamic Rekeying in 3GPP Multimedia Broadcast/Multicast Service (MBMS)
Jeng-Feng Weng, Student Member, IEEE, and Jyh-Cheng Chen, Senior Member, IEEE

Abstract—In multicast, it is important to update security keys when a user joins or leaves the multicast group. In this paper, we study how to reduce the cost to update the security keys in 3GPP MBMS. The problem is especially critical for MBMS because 3G is a large scale network with huge number of mobile users. We show that with dynamic rekeying, the cost can be reduced significantly.

Index Terms—Key management, 3GPP MBMS, multicast, security, wireless networks.

I. INTRODUCTION

The 3GPP has defined Multimedia Broadcast/Multicast Service (MBMS) [1]–[3]. To prevent unauthorized users from accessing the multicast contents, the security key(s) used for multicast transmission should be updated when a user leaves the multicast group. The security keys need to be updated when a new user joins the multicast group as well so that the new user can not decrypt the multicast messages sent before the user joins the multicast group. As users may join or leave the multicast group frequently in wireless networks, it is critical to reduce the rekeying cost. Because 3G is a large scale network with huge number of users, it is a challenge to perform rekeying efficiently.

Studies have shown that Logical Key Hierarchy (LKH) with dynamic rekeying can reduce the cost in multicast [4]. Without dynamic rekeying, when the number of users increases, the rekeying cost will increase rapidly. In this paper, we demonstrate that by performing dynamic rekeying in a proper constructed key tree, the rekeying cost can be reduced significantly. Based on the derivations in [4], we compare the costs with and without dynamic rekeying. We show that when key tree has degree of 4, the performance is the best. In this paper, we quantify the rekeying cost with large number of users. The results are valuable in large scale deployment of commercial 3GPP MBMS.

II. BACKGROUND

The LKH is widely adopted in key management of IP multicast [4]–[6]. It is a tree with single root and two parameters: (1) height, $h$, which is the longest path from a leaf to the root, and (2) degree, $d$, which is the maximum number of outgoing edges of a node in the tree. The tree is called key tree. Each leaf in the key tree represents a unique user. Every user owns three keys: (1) individual key which is shared with the Key Server (KS), (2) group key which is shared with the KS and all other users in the multicast group, and (3) auxiliary key which is stored in an intermediate node, the user, and the KS. For example, in Fig. 1(b), $u_1$ is a leaf that represents User 1. It has an individual key $K_1$. The $K_{1-9}$ is a Session Encryption Key (SEK) that essentially is the group key for the multicast group. It is shared among $u_1, \ldots, u_9$. The KS creates and distributes the SEK to the authorized users in the multicast group. When a user leaves the multicast group, the SEK must be updated to prevent eavesdropping from the past member of the multicast group. The $K_{123}$ is a Key Encryption Key (KEK), that is shared among $u_1, u_2, \text{and} u_3$. It is an auxiliary key. The purpose of KEK will be described later. Each user in Fig. 1(b) holds 3 keys, for example, $K_{1}, K_{123}$, and $K_{1-9}$ for $u_1$. The KS holds 13 keys ($K_{1-9}$, $K_{123}$, $K_{156}$, $K_{789}$, $K_{1, \ldots, K_9}$). In a full and balanced $d$-ary key tree with height $h$ and $n$ group
members, i.e., \( n = d^h - 1 \), based on the study in [4], each user holds \( h \) keys and the total number of keys held by the KS is:

\[
1 + d + d^2 + \ldots + d^{h-1} = (d^h - 1)/(d-1) \approx n(d/(d-1)) \tag{1}
\]

The key-oriented rekeying [4] is widely used as a key management technique in secure multicast. Here, we briefly describe the key-oriented rekeying.

We use the following notation:

\[
KS \rightarrow u : \{x\}_y
\]

to denote the sending of message \( \{x\}_y \) from KS to user \( u \). The notation \( \{x\}_y \) denotes that the key \( x \) is encrypted by using key \( y \).

In Fig. 1(a), when \( u_6 \) is permitted to join the multicast group by key-oriented rekeying, the KS needs to send the following rekeying messages so the new key tree will be like the one shown in Fig. 1(b):

\[
KS \rightarrow \{u_1, \ldots, u_8\} : \{K_{1-9}\}_K_{1-8}
\]
\[
KS \rightarrow u_9 : \{K_{1-9}\}_{K_9}
\]
\[
KS \rightarrow \{u_7, u_8\} : \{K_{789}\}_{K_{78}}
\]
\[
KS \rightarrow u_9 : \{K_{1-9}, K_{789}\}_{K_9}
\]

We can use the join protocol for a tree key graph with key-oriented rekeying in [4] to combine multiple rekeying messages into one. Therefore, the KS can send the combined rekeying message to particular users by multicast. This approach can reduce the number of rekeying messages. Thus, the KS only sends the following rekeying messages:

\[
KS \rightarrow \{u_1, \ldots, u_6\} : \{K_{1-9}\}_K_{1-8}
\]
\[
KS \rightarrow \{u_7, u_8\} : \{K_{789}\}_{K_{78}}
\]
\[
KS \rightarrow u_9 : \{K_{1-9}, K_{789}\}_{K_9}
\]

In Fig. 1(b), when user \( u_9 \) leaves the multicast group, by key-oriented rekeying using the leave protocol for a tree key graph in [4], the KS will send the following four rekeying messages so the key tree will become the one shown in Fig. 1(a):

\[
KS \rightarrow \{u_1, u_2, u_3\} : \{K_{1-8}\}_K_{123}
\]
\[
KS \rightarrow \{u_4, u_5, u_6\} : \{K_{1-8}\}_K_{456}
\]
\[
KS \rightarrow u_7 : \{K_{1-8}, K_{78}\}_K_{78}
\]
\[
KS \rightarrow u_8 : \{K_{1-8}, K_{789}\}_K_{8}
\]

The example illustrated in Fig. 1 demonstrates that with LKH, which creates a hierarchy key structure with KEKs in the intermediate nodes, the server only needs to perform 5 encryptions and send 4 rekeying messages when \( u_9 \) leaves the multicast group.

Fig. 2 illustrates an example without dynamic rekeying. There is no intermediate node. Assuming there are 9 users, i.e. \( n = 9 \) in Fig. 2, and the current SEK is \( K_{1-9} \). When \( u_9 \) leaves the multicast group, the KS needs to generate a new SEK, \( K_{1-8} \). The new SEK cannot be encrypted by \( K_{1-9} \) and sent by multicast. Otherwise, \( u_9 \) will know the new SEK. Thus, the KS needs to perform 8 encryptions (encrypt \( K_{1-8} \) by \( K_{1}, \ldots, K_8 \)) and send the 8 rekeying messages to \( u_1, \ldots, u_8 \) by unicast.

Comparing the examples shown in Fig. 1 and Fig. 2, we can see that LKH with dynamic rekeying can reduce the cost. As that in [4], we mainly consider the computational cost which is the number of key encryptions and decryptions required by a join/leave request. By using join and leave protocols for a key tree with key-oriented rekeying to each join/leave request, the average cost per request of the server and user are denoted by Equ. (2) and Equ. (3), respectively [4]:

\[
C_s = (d + 2)(h - 1)/2
\tag{2}
\]
\[
C_u = d/(d - 1)
\tag{3}
\]

There are four keys in MBMS: MBMS Request Key (MRK), MBMS User Key (MUK), MBMS Service Key (MSK), and MBMS Traffic Key (MTK). MRK is mainly used for authentication. MUK is used to protect the distribution of MSK. The main purpose of MSK is to protect a certain MBMS session. It is also used to protect the distribution of MTK. It, however, is not used to encrypt/decrypt MBMS traffic. The encryption and decryption is done by using MTK. MUK and MSK are delivered by using Multimedia Internet Keying (MIKEY) (IETF RFC 3830). Both Broadcast Multicast Service Center (BM-SC) and User Equipment (UE) own the four keys. Details can be found in [1]–[3].

In this section, we first derive the costs for MBMS without dynamic rekeying. We then discuss two LKH approaches with dynamic rekeying for MBMS. The computational cost and storage cost are compared and shown in Table I and Table II.

### A. MBMS without Dynamic Rekeying

In MBMS, the BM-SC and each UE share the same MRK, MUK, MSK, and MTK. Without dynamic rekeying, the key tree essentially looks like that illustrated in Fig. 2. The \( K_n \) in Fig. 2 represents the four keys held by user \( n (u_n) \). Because there is no KEKs in intermediate nodes, the height of the key tree is 2. When a user joins the multicast group, the degree of the key tree is increased. Based on Equ. (2) and Equ. (3), we can derive the computational cost when the number of users increases. The results are shown in Table I. For the storage cost, the KS (BM-SC) and all group members share the same MTK and MSK. The KS also shares the MUK and MRK with each individual user. Therefore, there are \( 2d + 2 \) keys in the KS and \( 4 \) keys in each user. Table II shows the storage cost, where \( K_u \) and \( K_s \) denote the numbers of keys stored in the KS and each user, respectively.

![Fig. 2. Key tree without dynamic rekeying.](image-url)
Table I

<table>
<thead>
<tr>
<th># of users</th>
<th>C_d</th>
<th>C_H3</th>
<th>C_d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^8</td>
<td>1.000015</td>
<td>32,769</td>
<td>1.000015</td>
</tr>
<tr>
<td>2^9</td>
<td>1.000015</td>
<td>32,769</td>
<td>1.000015</td>
</tr>
<tr>
<td>2^10</td>
<td>1.000015</td>
<td>32,769</td>
<td>1.000015</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th># of users</th>
<th>K_d</th>
<th>K_H3</th>
<th>K_d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^8</td>
<td>39,146,837</td>
<td>4,098</td>
<td>36</td>
</tr>
<tr>
<td>2^9</td>
<td>1,000024</td>
<td>8,257</td>
<td>8</td>
</tr>
<tr>
<td>2^10</td>
<td>1,000024</td>
<td>8,257</td>
<td>8</td>
</tr>
</tbody>
</table>

B. Dynamic Rekeying: LKH with Height 3

As aforementioned discussion, MRK is used for authentication rather than encryption. MTK is the group key which is used to protect the MBMS traffic. MTK is protected by MSK, which is further protected by MUK. Therefore, a natural way to support dynamic rekeying in MBMS is to construct the key tree as shown in Fig. 3, where MTK, MSK, MUK essentially are the group key, auxiliary key, and individual key discussed in Section II. The height of the key tree is fixed to 3. When a user joins the multicast group, the degree of a sub-tree is increased. Also based on Equ. (2) and Equ. (3), we can derive the computational cost, which is also shown in Table I. Each user holds 4 keys. Based on Equ. (1), we can derive that the KS holds \( d + 1 \) keys. The numerical results are also shown in Table II. To support this approach, some intermediate nodes in the MBMS network must hold MSKs.

C. Dynamic Rekeying: LKH with Degree 4

In [4], the authors prove that to minimize rekeying cost, the degree of the key tree should be set as 4 as that depicted in Fig. 4. This is the optimal solution. Based on Equ. (2) and Equ. (3), Table I shows that the computational cost is reduced significantly when the degree is fixed to 4. However, we need to increase the height of the key tree when the number of users increases. Therefore, we need to insert keys as KEKs in the intermediate nodes as that illustrated in Fig. 4. In Fig. 4, the \( K_i, i \in \{1, 2, 3, 4\} \), are the keys put in the children of the KS (BM-SC). The \( K_{1i}, i \in \{1, 2, 3, 4\} \), are the keys put in the children of node \( K_1 \). Each user holds \( h + 1 \) keys. Based on Equ. (1), there are \( d + 1 \) keys in the KS. Table II shows the storage cost. Similar to that in Section III-B, to support this approach, KEKs must be inserted in intermediate nodes in the MBMS network.

Table II shows that the storage cost is similar in the three different approaches. Although the storage cost in LKH with degree 4 is higher, the difference is minimal. The KS (BM-SC) usually has enough memory to accommodate the increase in memory shown in Table II. The computational cost plays a more important role.

IV. CONCLUSION

In this paper, we quantify the computational and storage costs of three different approaches for rekeying in MBMS. We demonstrate that without dynamic rekeying, the computational cost will increase rapidly when the number of users increases. It is not a scalable solution for a large scale network. Because 3G is a large scale network with huge number of users, it is suggested that LKH with dynamic rekeying should be deployed in MBMS.

REFERENCES