Key Management for UMTS MBMS

Shin-Ming Cheng, Member, IEEE, Wei-Ru Lai, Member, IEEE, Phone Lin, Senior Member, IEEE, and Kwang-Cheng Chen, Fellow, IEEE

Abstract—3GPP 33.246 proposes Key Management Mechanism (KMM) to distribute security keys for Universal Mobile Telecommunications System (UMTS) Multimedia Broadcast and Multicast Service (MBMS). KMM introduces extra communication overhead to UMTS. The previous study, Key-Tree Scheme (KTS), resolves this issue for the IP multicast network. However, this scheme may not be so efficient while applied in UMTS MBMS due to lots of storage space and heavy multicast traffic introduced, which may decrease the QoS of UMTS MBMS. In this paper, we propose a more efficient scheme, Hash Function Scheme (HFS), to release both storage and communication overhead for KMM in UMTS MBMS. We first modify the KTS applied in the UMTS MBMS and then detail the execution of HFS, which is proven to be correct. We conduct an analytical model and simulation experiments to compare the performance between the UMTS KMM with KTS and with HFS. Our study shows that the proposed HFS can reduce both communication and storage overhead without damaging QoS of UMTS MBMS.

Index Terms—Hash function, key management, multimedia broadcast and multicast service (MBMS), Universal Mobile Telecommunications System (UMTS).

I. INTRODUCTION

TO DELIVER multimedia content efficiently over the Universal Mobile Telecommunications System (UMTS), 3GPP proposed the Multimedia Broadcast/Multicast Service (MBMS) based on UMTS [1], [2]. UMTS MBMS utilizes point-to-multipoint transmission technology, where the multimedia content is delivered from a single source to a group of mobile devices through the UMTS MBMS transmission bearer.

Figure 1 illustrates the simplified UMTS MBMS network architecture [3], without showing the network elements supporting MBMS transmission bearers in UMTS. The User Equipment (UE; Figure 1 (a)) receives the MBMS application (also known as MBMS User Service) [4] from the Broadcast-Multicast Service Center (BM-SC; Figure 1 (b)), which is an application server serving as an MBMS data source or as an entry point for the multimedia content provider. The UEs joining the multicast group for a specific MBMS User Service are called joined UEs. The BM-SC initializes the establishment of the MBMS transmission bearer, then sends multimedia content to the joined UEs. The Home Subscriber Subsystem (HSS; Figure 1 (c)) maintains UMTS subscriber information (e.g., security-related information). The Bootstrapping Server Function (BSF; Figure 1 (d)) is a security server function, which is responsible for establishing shared secrets between the BM-SC and UEs.

The BM-SC multicasts MBMS content to the joined UEs via a broadcasting network bearer, where the MBMS point-to-multipoint Traffic Channel (MTCH) in the air interface [5] is used to carry the multicast content. To prevent the non-joined UEs from receiving the MBMS content, 3GPP 33.246 proposed the Key Management Mechanism (KMM) [6], which are described in detail below. A specific MBMS User Service has two corresponding group keys, namely the MBMS Transmission Key (MTK; denoted as $T$) and the MBMS Service Key (MSK; denoted as $S$). Every UE of an MBMS User Service group has the same $S$ and $T$. $T$ is used to protect multicast content from eavesdropping or modification, where the multicast content is encrypted by $T$ before being multicast to all joined UEs. A UE uses $T$ to decrypt content that it receives. $T$ is multicast from BM-SC to all joined UEs by sending $S[T]$, which means that $T$ is encrypted by $S$. $S$ is individually unicasted from BM-SC to every joined UE.

During an MBMS User Service, $T$ or $S$ is updated when one of the following events occurs: (Event 1) a new UE joins the multicast group; (Event 2) a joined UE leaves the multicast group; (Event 3) the timer of the current $S$ expires, or (Event 4) the timer of the current $T$ expires. The User Service Join procedure (denoted as $P_1$ for Event 1), the User Service Leave procedure (denoted as $P_2$ for Event 2), the MSK Periodic Update procedure (denoted as $P_3$ for Event 3), and the MTK...
Periodic Update procedure (denoted as \(P_4\) for Event 4) are exercised at this moment in order to update \(T\) or \(S\) [6]. The four procedures are described in detail below.

Figure 2 shows the message flow for Procedure \(P_1\) with the following steps, where we suppose that the multicast group contains \(N\) joined UEs. Assume that a new UE, UE\(_{N+1}\), joins the MBMS User Service, and before UE\(_{N+1}\) joins the service, the two keys, \(S^{\text{old}}\) and \(T^{\text{old}}\) are used for the MBMS User Service.

**User Service Join Procedure \(P_1\):**

**Step J1.** UE\(_{N+1}\) performs the bootstrapping authentication procedure [3] with BSF to obtain an MBMS Request Key (denoted as \(R_{N+1}\)) and an MBMS User Key (denoted as \(U_{N+1}\)).

**Step J2.** UE\(_{N+1}\) uses \(R_{N+1}\) as the authentication password when executing the bootstrapping usage procedure [3] with BM-SC and BSF.

**Step J3.** If the authentication in Step J2 is successful, then BM-SC generates \(S^{\text{new}}\), and unicasts it to every UE, UE\(_i\), in the multicast group by sending \(U_i\{S^{\text{new}}\}\). Otherwise (i.e., the authentication fails), the procedure quits. This step requires \(N+1\) unicasts to deliver \(S^{\text{new}}\).

**Step J4.** BM-SC generates \(T^{\text{new}}\), and multicasts it to all joined UEs by sending \(S^{\text{new}}\{T^{\text{new}}\}\). Significantly, only one multicast transmission is necessary.

The other three procedures are similar to Procedure \(P_1\). Procedure \(P_2\) consists of three steps, Steps L1–L3, which are the same as Steps J2–J4, respectively. Procedure \(P_3\) comprises two steps, Steps S1 and S2, which are the same as Steps J3 and J4, respectively. Procedure \(P_4\) consists of only one step, Step T1, which is the same as Step J4.

Note that in Step J3, \(S\) is unicasted through the Dedicated Control Channel (DCCH), which is a signaling message. Conversely, in Step J4, \(T\) is multicasted using the MIKEY protocol [7], and \(T\) is delivered via the MTCH, which is used to carry the multimedia content and other session information. In other words, \(T\) delivery may consume the radio resource for the transmission of multicast content. This study considers the following two main issues.

**Issue 1.** The number of unicast key deliveries should be minimized to lower signaling overhead.

**Issue 2.** The number of multicast key deliveries should be minimized to provide an acceptable QoS (i.e., more bandwidth can be used to transmit multicast content).

Only one group key (that is used for data encryption and unicast to every member of a multicast group) is defined in IP multicast networks. Previous studies [8]–[11] have attempted to reduce the number of unicasting for the group key deliveries in IP multicast networks by proposing Key-Tree Scheme (KTS), which applies multicast Key Encryption Keys (KEKs; cf. Section II) to all members of a multicast group. The KTS was applied in cellular networks in 2004 [12], [13], when the UMTS MBMS has not been well defined (i.e., only one group key was considered in these studies).

In this work, to consider Issues 1 and 2, KTS is first modified so that it can be applied in UMTS MBMS KMM. Analytical results indicate that KTS is not efficient in UMTS MBMS KMM. The Hash Function Scheme (HFS), which is regarded as more efficient than KTS, is then proposed. The rest of this paper is organized as follows. The application of KTS in the existing UMTS MBMS KMM is described in Section II. Section III details HFS. Section IV provides security analysis for HFS. Section V conducts an analytical model and simulation experiments to evaluate the performance of KMM with/without KTS or HFS. Finally, Section VI concludes this work.

## II. KTS in UMTS MBMS Key Management

This section describes how to apply KTS in UMTS MBMS KMM. In KTS, BM-SC establishes and maintains a balanced binary key tree [8], [9]. As shown in Figure 3, each leaf \(U\) of the tree is assigned to corresponding joined UE (Figure 3 (a)). The root of the key tree is \(S\) for the multicast group (Figure 3 (b)). The intermediate nodes of the key tree are the intermediate KEKs (Figure 3 (c)), which are used to facilitate efficient \(S\) updates.

Consider \(N\) joined UEs, UE\(_1\), UE\(_2\), ..., UE\(_N\), in the multicast group. Let \(H\) be the height of the binary tree, which...
can be calculated by $H = \lfloor \log N \rfloor$. The keys in the tree have the index number $(i, j)$, where $0 \leq i < H$ is the layer number, and $1 \leq j \leq 2^i$ is the position number in layer $i$. The index number for the parent of the KEK with the index $(i, j)$ is given by $(i - 1, \lceil \frac{j}{2} \rceil)$. Suppose that UE$_i$ is assigned the user key $U_{H,j}$ where $1 \leq j \leq N$. The content is encrypted by the intermediate key $K_{H,j}$ before it is multicasted to $2^{H-i}$ UEs, $U_{2^{H_1} - (j-1)+1}$, ..., $U_{2^{H_1} - j}$. $U_{H,j}$ is used to encrypt the key that will be unicast to UE$_j$. UE$_j$ stores $S$, $T$, $R_j$, $U_{H,j}$ and $H-1$ intermediate keys, $K_{H-1,1}$, $K_{H-2,1}$, ..., $K_{1,1}$. In other words, UE$_j$ contains $H + 3$ keys.

In the original KMM in UMTS, the new $S$ should be unicasted to all joined UEs to update an old $S$. In KTS, the multicast technology can be applied to deliver the new $S$. Consider Figure 3 as an example. To deliver a new $S$ to UE$_1$, UE$_2$, ..., UE$_8$, BM-SC can multicast $K_{1,1}(S_{\text{new}})$ to UE$_1$, UE$_2$, UE$_3$, and UE$_4$ and multicast $K_{1,2}(S_{\text{new}})$ to UE$_5$, UE$_6$, UE$_7$, and UE$_8$. To apply KTS in KMM, Procedure P4 is not modified, while the other three procedures are modified as follows:

**User Service Leave Procedure** P2: The Steps L1 and L3 are the same as those in KMM while the Step L2 is modified as follows. Assume that UE$_j$ leaves the multicast group. The group keys (including $S$ and $H-1$ KEKs) known by UE$_j$ should be updated so that UE$_j$ cannot decode any future multicast content. $K_{\text{old}}$ is updated to $K_{\text{new}}$, $K_{\text{old}}$ is updated to $K_{\text{new}}$, $K_{\text{old}}$ is updated to $K_{\text{new}}$, and $S_{\text{old}}$ is updated to $S_{\text{new}}$. All newly generated keys should be delivered to all joined UEs that own the old keys. The following actions are taken. The KEK, $K_{\text{new}}$, is unicasted to the other UE that owns $K_{\text{old}}$ (i.e., UE$_{i+1}$ if $i$ is odd, or UE$_{i-1}$ if $i$ is even). $K_{\text{new}}$, $K_{\text{new}}$, and $S_{\text{new}}$ are multicast to the UEs that own the old keys, and are encrypted with each of their respective children’s KEKs. Take Figure 3 as an example. Originally, there are 7 joined UEs, UE$_1$, UE$_2$, ..., UE$_7$, and later UE$_8$ leaves the multicast group. In this case, the following four steps are exercised in Step L2.

**Step L2.1.** The BM-SC updates the two old KEKs, $K_{1,2}^{\text{old}}$ and $K_{2,3}^{\text{old}}$ as $K_{1,2}^{\text{new}}$ and $K_{2,3}^{\text{new}}$, and the old MSK $S_{\text{old}}$ as $S_{\text{new}}$ for UE$_5$ by unicasting $U_{3,5}(K_{1,2}^{\text{new}}, K_{2,3}^{\text{new}}, S_{\text{new}})$ to UE$_5$.

**Step L2.2.** The BM-SC updates the old KEK $K_{1,2}^{\text{old}}$ as $K_{1,2}^{\text{new}}$ for UE$_7$ by multicasting $K_{1,2}^{\text{new}}$ to UE$_7$.

**Step L2.3.** The BM-SC updates the old MSK $S_{\text{old}}$ as $S_{\text{new}}$ for UE$_1$, UE$_2$, ..., UE$_4$ by multicasting $S_{\text{new}}$ to UE$_1$, UE$_2$, UE$_3$, and UE$_4$.

**Step L2.4.** The BM-SC updates the old MSK $S_{\text{old}}$ as $S_{\text{new}}$ for UE$_5$ and UE$_7$ by multicasting $S_{\text{new}}$ to UE$_5$ and UE$_7$.

Note that the key tree may not be balanced when a UE leaves. As recommended by Moyer et al. [14], the key tree should be regenerated by running the Re-balance algorithm. After the key tree regeneration, the newly generated keys should be delivered to the affected joined UEs. As noted in [14], the number of keys that need to be updated is twice that in a non-balanced key tree after a UE leaves.

**User Service Join Procedure** P1: When a new UE, UE’, joins the multicast group, the BM-SC first determines the corresponding position in the key tree for UE’ by executing the Re-balance algorithm in [14]. Let $k$ be the position number of the found position, i.e., UE’ is assigned $U_{H,k}$. To simplify our description, UE’ is denoted as $U_{k}$ hereafter.

To prevent UE$_k$ from decoding overheard multicast content, $K_{\text{old}}$, $K_{\text{old}}$, $K_{\text{old}}$, and $S_{\text{old}}$ should be updated. The newly generated keys (i.e., $K_{\text{new}}$, $K_{\text{new}}$, $K_{\text{new}}$, and $S_{\text{new}}$) are delivered to all joined UEs that own the old keys, which are encrypted by the old keys. BM-SC then unicasts $U_{H,k}(K_{\text{new}}, K_{\text{new}}, K_{\text{new}}, S_{\text{new}})$ to UE$_k$. In the example of Figure 3, when UE$_k$ joins the multicast group, BM-SC multicasts $K_{1,2}^{\text{old}}$ to UE$_5$, UE$_6$, and UE$_7$, and unicasts $U_{5,8}(K_{1,2}^{\text{new}}, K_{1,2}^{\text{new}}, S_{\text{new}})$ to UE$_8$, in order to deliver $K_{1,2}^{\text{new}}$ to UE$_5$, UE$_6$, UE$_7$, and UE$_8$. These key deliveries are performed at Step J3.

**MSK Update Procedure** P3: To update $S$, the all KEKs and $S$ in the key tree should be regenerated and unicasted to all joined UEs, including $S$ and $H-1$ KEKs. The key deliveries can be performed at Step S1.

In KTS, delivery of intermediate KEKs requires multicast transmission. According to the UMTS KMM, KEKs may be delivered through the MTCH. Based on the UMTS MBMS standard [4], the following two implementation methods are available for KEK delivery: (i) BM-SC creates a new multicast group for the KEK delivery, and (ii) BM-SC multicasts KEKs through the network bearer of the original multicast group.

In method (i), to form a new multicast group, all joined UEs should perform the MBMS Multicast Service Activation procedure [2], which incurs heavy signaling overhead to the UMTS network. Method (ii) is thus more practical than method (i). However, method (ii) consumes radio resource (carrying the multicast content) in delivering KEKs, thus decreasing the QoS of multicast content. Furthermore, KTS has the following problems.

- In KTS, $H + 3$ keys are stored in a UE. Increasing the number of joined UEs (i.e., increasing $H$) raises the amount of storage space required, and therefore may not be practical due to the limited UE storage space.
- KTS may require much extra key transmission overhead to keep the tree balanced when UEs join or leave.

The next section proposes the Hash Function Scheme (HFS) for KMM in UMTS MBMS by utilizing the one-way hash function to resolve both Issues 1 and 2 without extra storage space.

### III. Hash Function Scheme

A one-way hash function $h(\cdot)$ is a powerful and computationally efficient cryptographic tool [15], which takes a message of arbitrary size as its input, and outputs a fixed string. “One way” means that the original input cannot feasibly...
be derived from the output. The one-way property of hash function is utilized to update $S$ efficiently. The idea of HFS is that BM-SC requests (through multicast) UEs to generate a new $S$ by using $h(\cdot)$ instead of unicast $S$ to all UEs. The HFS exercises as follows. Suppose that the multicast group contains $N$ joined UEs, namely $U_{E1}$, $U_{E2}$, ..., $U_{EN}$, and $S^{old}$ and $T^{old}$ are used for the MBMS User Service. To apply HFS to KMM, Procedures P1 and P3 are modified as follows, while the other two procedures (P2 and P4) remain the same as those in KMM.

**User Service Join Procedure** P1: Figure 4 shows the message flow for this procedure, where Steps J1 and J2 are the same as that in KMM, and Steps J3 and J4 are modified. Assume that a UE, $U_{EN+1}$, joins the multicast group. If $N = 0$ (i.e., $U_{EN+1}$ is the only user in the multicast group), then this procedure is the same as that in KMM of MBMS UMTS. For $N > 0$, $U_{EN+1}$ is assigned $U_{EN+1}$ after being successfully authenticated. The BM-SC generates a new $T$, $T^{new}$ and a new $S$ by executing $S^{new} = h(T^{new}, S^{old})$. Then BM-SC unicasts $U_{N+1}(S^{new}, T^{new})$ to $U_{EN+1}$. Then BM-SC multisects $S^{old}(T^{new})$ to the other $N$ joined UEs. The $N$ UEs generate $S^{new}$ by executing $S^{new} = h(T^{new}, S^{old})$ respectively.

**MSK Update Procedure** P3: The BM-SC generates a new $T$, $T^{new}$ and a new $S$ by executing $S^{new} = h(T^{new}, S^{old})$. Then, BM-SC multisects $S^{old}(T^{new})$ to $N$ joined UEs. The $N$ UEs generate $S^{new}$ by executing $S^{new} = h(T^{new}, S^{old})$, respectively.

The SHA-1 [16] (the standard one-way hash function installed in the UE) can be utilized to implement HFS. The implementation cost of HFS is considered insignificant. For the robustness of SHA-1, as mentioned in [15], theoretically, it requires $2^{80}$ trials using the brute-force method to break the full 80-step SHA-1, which is considered big overhead. In the recent studies [15], [17], the birthday attack and multicollision attack were proposed to break SHA with less computation overhead, whose details can be found in [15], [17]. Wang et al. [18] reduced the complexity of the computation (to find a collision in SHA-1 using collision search attack) to $2^{69}$. The computational overhead is still high, i.e., up to several hours. In HFS, the one-way hash function $h(\cdot)$ is applied when only Event 1 or 3 occurs. For Events 2 and 4, HFS follows the standard procedures in MBMS KMM. Usually, the time interval between the occurrence of Event 2 and the occurrence of Event 4 is shorter than one hour. In other words, before SHA-1 is broken, UE may retrieve new $S$ and $T$ from BM-SC. Thus, HFS is considered robust under birthday and multicollision attacks.

**IV. Security Analysis**

A secured multicast mechanism should satisfy the group secrecy property [19], which stipulates the following requirements.

- **Nongroup confidentiality**: only the joined UEs can decode the multicast content, i.e., non-joined UEs cannot decode it.
- **Forward secrecy**: a UE joining at time $t$ cannot decode any multicast content before $t$.
- **Backward secrecy**: a UE leaving at time $t$ cannot decode any multicast content after $t$.

This section analyzes the group secrecy property for the KMM, KMM with KTS (denoted as KMM$_{KTS}$), and KMM with HFS (denoted as KMM$_{HFS}$).

As specified in [6], in KMM, nongroup confidentiality can be achieved by group keys $S$ and $T$, and forward and backward confidentialities can be achieved via Procedures P1 and P2, respectively. Additionally, in [8], KMM$_{KTS}$ has been proven to be able to achieve the three confidentialities. In KMM$_{HFS}$, Procedure P2 is the same as that in KMM, and backward secrecy can be achieved. In KMM$_{HFS}$, we modify Procedures P1 and P3 in KMM. In KMM$_{HFS}$, Procedure P1 is invoked to update $S$ and $T$ when a new UE joins the multicast group at $t$. Since the UE does not have the old $T$ and $S$, it cannot decode any content multicast before $t$, and forward secrecy holds in KMM$_{HFS}$.

In KMM, $T$ is used to encode the multicast content for security protection, and $S$ is used to encrypt the multicast transmission of $T$. The following lemma proves that HFS prevents any malicious UE from obtaining $S$ and $T$, and therefore cannot steal the multicast content. In other words, KMM$_{HFS}$ holds nongroup confidentiality.

**Lemma 1**: Let $t_i$ be the time when the $i$th event occurs during a multicast session, and $S^{(i)}$ and $T^{(i)}$ denote $S$ and $T$ used at the $i$th event. Suppose that a malicious UE, $U_{Em}$, starts to overhear the multicast information at time $t'_i$ during the period between $t_i$ and $t_{i+1}$, i.e., $t_i \leq t'_i < t_{i+1}$. Then with KMM$_{HFS}$, $U_{Em}$ cannot get $S^{(i)}$ and $T^{(i)}$.

**Proof**: The proof is completed by considering the following two conditions.

**Condition 1**: $t'_i > t_i$. The multicast information (overheard by $U_{Em}$ during the time period $[t'_i, t_{i+1})$) is $T^{(i)} \{\text{content}\}$, and $U_{Em}$ cannot retrieve $S^{(i)}$ and $T^{(i)}$ from this information.

**Condition 2**: $t'_i = t_i$. During the time period $[t_i, t_{i+1})$, $U_{Em}$ can overhear $S^{(i)} \{T^{(i)}\}$ and $T^{(i)} \{\text{content}\}$. In this condition, if $U_{Em}$ cannot get $S^{(i)}$, then he cannot steal the content. Hypothesis “$U_{Em}$ cannot get $S^{(i)}$” is proven to hold by induction on $i$.

**Basic**: If $i = 1$ in KMM, then the $i$th event must be Event 1. The first UE joins the multicast group, and $S^{(1)}$ is unicasted with protection to this UE. The $U_{Em}$ cannot obtain $S^{(1)}$, and the hypothesis holds.

**Inductive Step**: Suppose that the hypothesis holds when $i = k$ (i.e., $U_{Em}$ cannot get $S^{(k)}$). For $i = k + 1$, consider the following four cases:

**Case 1**: The $k + 1$st event is Event 1. At $t_{k+1}$, all joined UEs respectively generate $S^{(k+1)}$ by executing Procedure P1 in KMM$_{HFS}$, and have

$$S^{(k+1)} = h(T^{(k+1)}, S^{(k)})$$

where $T^{(k+1)}$ is delivered by multicast $S^{(k)} \{T^{(k+1)}\}$. Since $U_{Em}$ cannot obtain $S^{(k)}$, he cannot retrieve $S^{(k+1)}$.
Fig. 4. Message flow for the User Service Join procedure in HFS.

Case 2: The k + 1st event is Event 2. At \( t_{k+1} \), \( S^{(k+1)} \) is unicasted with protection to all joined UEs by BM-SC (see Procedure P2 in KMM\_HFS), and thus UE\(_{m} \) cannot get \( S^{(k+1)} \).

Case 3: The k + 1st event is Event 3. At \( t_{k+1} \), all joined UEs respectively generate \( S^{(k+1)} \) by executing Procedure P3 in KMM\_HFS using (1). In the same reason mentioned in Case 1, UE\(_{m} \) cannot get \( S^{(k+1)} \).

Case 4: The k + 1st event is Event 4. At \( t_{k+1} \), all joined UEs update \( T \) by receiving the multicasted \( S^{(k+1)} \{ T^{(k+1)} \} \) from BM-SC (see Procedure P4 in KMM\_HFS). \( S^{(k+1)} \) is the same as \( S^{(k)} \). Since UE\(_{m} \) cannot obtain \( S^{(k)} \), \( S^{(k+1)} \) cannot be retrieved by UE\(_{m} \).

Thus, the hypothesis holds for all cases.

V. PERFORMANCE EVALUATION

This study develops an analytic model and simulation experiments to investigate the performance for KMM, KMM\_KTS and KMM\_HFS. Appendix A describes the analytic model in details. The simulation model adopts the event-driven approach widely used in mobile network studies [20]. Both models are validated against each other in Appendix A. More than 500,000 experiments are executed to ensure the stability of the simulation results.

In this study, the session time \( T_{s} \) of an MBMS User Service session is modeled as a widely used Gamma distribution with mean \( \frac{1}{\mu} \) and variance \( \nu_{s} \) [20] because Gamma distribution can be shaped to represent many distributions, as well to as measure data that cannot be characterized by a particular distribution [21]. Additionally, this distribution has been widely used in mobile network studies [20]. A UE may join or leave the multicast group during an MBMS User Service session. Assume that the UE inter-arrival time \( t_{ua} \) to an MBMS User Service session has Gamma distribution with mean \( \frac{1}{\mu} \) and variance \( \nu_{ua} \). The resident time \( t_{ru} \) (when the joined UE stays in an MBMS User Service session) is assumed to be Gamma distributed with mean \( \frac{1}{\mu} \) and variance \( \nu_{ru} = \frac{1}{\alpha^{2}\mu^{2}} \), where \( \alpha \) is the shape parameter. During an MBMS User Service session, \( S \) and \( T \) are periodically updated every \( \Delta_{s} \) and \( \Delta_{t} \) time units, respectively. The other notations used in this study include the random number \( N \) (to count the number of current joined UEs in a multicast group at a specific time point), the total number \( X \) of UEs that have joined the multicast group during the period \( T_{s} \) and the total number \( Y \) of UEs that have left the multicast group during the period \( T_{s} \).

This study applies videos as multicast sessions to investigate the performance of MBMS User Services. The average session time \( (\frac{1}{\mu}) \) is 100 mins, and the variance of session time \( (\nu_{s}) \) is 570 min\(^2\). If \( \nu_{s} \) is large (e.g., \( \nu_{s} \geq \frac{1}{\mu} \)), then most of generated session times are very small numbers (the curve of Gamma density function can be found in Figure 11 of Chapter 3 [21]) and not fit to represent the periods of movies. Therefore, a small variance is chosen to approximate the real world. According to Chebyshev’s Inequality, the probability \( P \) that the session times are out of the range \( [\frac{1}{\mu} - t, \frac{1}{\mu} + t] \) is less than \( \frac{1}{t^{2}} \) for all \( t \). If \( \frac{1}{\mu} = 100 \) mins and \( \nu_{s} = 570 \) min\(^2\), then \( P \) is less than 63.3%. The numerical results indicate that \( P \) is about 20%. Furthermore, the effects of \( \frac{1}{\mu} \), \( \nu_{s} \) and \( \frac{1}{\eta} \) (discussed later) are similar even for different \( \nu_{s} \). Therefore, the performance results specifically for Exponential session times are not presented. For the same reason, \( a = 4 \) is chosen for the Gamma resident time \( t_{ru} \) so that a reasonable number of UEs stay in the MBMS service within any time period. Anyway, it is easy to adjust these parameters to simulate different behaviors of MBMS services.

Let \( e_{i} \) be the number of Events \( i (i = 1, 2, 3 \) or 4) occurring in a multicast session. Let \( N_{i,j} \) be the number of joined UEs in the multicast group at the \( j \)th occurrence of Event \( i \). As specified in [6], \( T \) and \( S \) have the same size of 128 bits. Let \( n_{u,i}(N_{i,j}) \) and \( n_{m,i}(N_{i,j}) \), be the numbers of unicast and multicast messages used to deliver \( S \) or \( T \) at the \( j \)th occurrence of Event \( i \) conditioning on there are \( N_{i,j} \) joined UEs, respectively. Let \( k_{u,i}(N_{i,j}) \) and \( k_{m,i}(N_{i,j}) \) be the numbers of keys (\( T \) or \( S \)) carried in a unicast and a multicast messages at the \( j \)th occurrence of Event \( i \) conditioning on there are \( N_{i,j} \) joined UEs, respectively. Let \( s_{u,i}(N_{i,j}) \) and \( s_{m,i}(N_{i,j}) \) be the total number of keys (\( T \) or \( S \)) carried in unicast and multicast messages at the \( j \)th occurrence of Event \( i \) conditioning on there are \( N_{i,j} \) joined UEs, respectively. For \( i = 1, 2, 3, 4 \), we have \( s_{u,i}(N_{i,j}) = n_{u,i}(N_{i,j})k_{u,i}(N_{i,j}) \) and \( s_{m,i}(N_{i,j}) = n_{m,i}(N_{i,j})k_{m,i}(N_{i,j}) \). Figure 5 lists the \( n_{u,i}(N_{i,j}), n_{m,i}(N_{i,j}), k_{u,i}(N_{i,j}), k_{m,i}(N_{i,j}), s_{u,i}(N_{i,j}), s_{m,i}(N_{i,j}) \) values for KMM, KMM\_KTS, and KMM\_HFS, where \( i = 1, 2, 3, 4 \). Note that the analysis for KMM\_KTS in Figure 5 is simplified due to the complexity of key tree regeneration. This study investigates the following output measures.
(2), (3) and Figure 5 clearly indicate that $M_m$ or $M_u$: the numbers of unicast or multicast messages used to deliver $S$ or $T$ during a multicast session, which is calculated as

$$M_u = \sum_{i=1}^{4} \sum_{j=1}^{e_i} n_{u,i(N_i,j)} \text{ or } M_m = \sum_{i=1}^{4} \sum_{j=1}^{e_i} n_{m,i(N_i,j)}. \quad (2)$$

- $S_u$ or $S_m$: the total number of keys ($S$ or $T$) carried in unicast or multicast messages during a multicast session, which is obtained as follows

$$S_u = \sum_{i=1}^{4} \sum_{j=1}^{e_i} s_{u,i(N_i,j)} \text{ or } S_m = \sum_{i=1}^{4} \sum_{j=1}^{e_i} s_{m,i(N_i,j)}. \quad (3)$$

(2), (3) and Figure 5 clearly indicate that $M_m = S_m$ for each scheme, since every multicast message contains only one key, i.e., $k_{m,1}(N_i,j) = k_{m,2}(N_i,j) = k_{m,3}(N_i,j) = k_{m,4}(N_i,j) = 1$ for all $j$. Additionally, KMM and KMM$_{HFS}$ have the same $M_m$ values. Based on the simulation experiments, the impacts of the input parameters on the output measures are elaborated as follows.

Effects of $\frac{1}{\lambda}$: Figure 6 plots $M_u$, $S_u$ and $S_m$ as functions of $\frac{1}{\lambda}$ for KMM, KMM$_{KTS}$ and KMM$_{HFS}$, where $\frac{1}{\mu} = 100$ mins, $\frac{1}{\eta} = 60$ mins. $\alpha = 4$, $\Delta_t = 5$ mins and $\Delta_s = 20$ mins. The inter-arrival time is exponentially distributed with mean $\frac{1}{\lambda}$ mins.

The figure shows an intuitive result that $M_u$, $S_u$ and $S_m$ decrease as $\frac{1}{\lambda}$ increases. A lower $\frac{1}{\lambda}$ implies that more UEs join (i.e., a larger $X$), stay (i.e., a larger $Y$) and then leave (i.e., a larger $Y$) the service. Based on Figure 5, explicitly all of $M_u$, $S_u$ and $S_m$ are decreasing functions when $\frac{1}{\lambda}$ increases.

Figure 6 (a) indicates that decreasing $\frac{1}{\lambda}$ significantly increases $M_u$ of KMM and slightly increases $M_u$ of KMM$_{HFS}$. Decreasing $\frac{1}{\lambda}$ only insignificantly affects $M_u$ of KMM$_{KTS}$. In KMM, BM-SC needs to unicast $S$ to all UEs in service at Procedures $P_1$, $P_2$ and $P_3$, and the effort of $S$ delivery is large if the net traffic is heavy. In KMM$_{HFS}$, $S$ is delivered to all UEs at Procedure $P_2$, and thus the $M_u$ of KMM$_{HFS}$ is smaller than that of KMM. In KMM$_{KTS}$, only for Procedure $P_3$, BM-SC uncasts $S$ to all UEs while $P_3$ rarely executes. Thus, $M_u$ of KMM$_{KTS}$ is insensitive to the change in $\frac{1}{\lambda}$, and is the smallest among the three schemes.

Since the wireless bandwidth is precious, the total number of keys carried in unicast/multicast messages is calculated to find the bandwidth consumed for each of the three schemes. Figure 6 (b) shows the total number of keys carried in unicast messages for the three schemes. The figure indicates when the traffic is small (i.e., $\frac{1}{\lambda} \geq 1.4$ mins), the $S_u$ of KMM$_{HFS}$ is less than that of KMM$_{KTS}$. Conversely, when the traffic is large (i.e., $\frac{1}{\lambda} < 1.4$ mins), KMM$_{KTS}$ requires fewer keys to deliver than KMM$_{HFS}$. As $\frac{1}{\lambda}$ decreases, $N$ and $Y$ increase, and therefore $S_u$ of KMM$_{HFS}$ increases rapidly since the number of unicast messages for Procedure $P_2$ is $O(N)$. For KMM$_{KTS}$, the keys of each unicast message at $P_1$, $P_2$ and $P_3$ is $O(\lceil \log N \rceil)$ and $S_u$ of KMM$_{KTS}$ increases slowly. If $N$ is small, then KMM$_{HFS}$ has a smaller $S_u$ than KMM$_{KTS}$.

When the net traffic is heavy, the gap between $O(N)$ and $O(\lceil \log N \rceil)$ increases rapidly, and KMM$_{KTS}$ performs better than the other two schemes.

Figure 6 (c) shows that as $\frac{1}{\lambda}$ decreases, $S_m$ of KMM$_{KTS}$ increases more rapidly than that of KMM and KMM$_{HFS}$. Note that $S_m = M_m$ for each scheme individually, and $S_m$ are the same for KMM and KMM$_{HFS}$. Since the number of keys carried in each multicast message at Procedures $P_1$ and $P_2$ is $O(\lceil \log N \rceil)$, $S_m$ of KMM$_{KTS}$ increases rapidly as $\frac{1}{\lambda}$ decreases. All of the multicast messages for KMM$_{KTS}$ are used for key distribution, and must be transmitted in real-time. Consider a special scenario in which many UEs join or depart simultaneously during a short period. Since $P_1$ and $P_2$ need to transmit many keys, and occupy the MTCH, the multimedia content is likely to be compressed, thus degrading QoS. Although KMM$_{KTS}$ has better performance for Issue 1, the requirement of Issue 2 is not guaranteed.

Effects of $\nu_a$: Figure 7 plot $S_u$ and $S_m$ as functions of $\nu_a$ for KMM, KMM$_{HFS}$ and KMM$_{KTS}$, respectively. In these figures, $\frac{1}{\mu} = 2$ mins, $\frac{1}{\eta} = 100$ mins, $\nu_a = 570$ min$^2$, $\frac{1}{\eta} = 4$, $\Delta_t = 5$ mins and $\Delta_s = 20$ mins. Note that if the variance ($\nu_a$) equals to 4 min$^2$, then the inter-arrival times form an Exponential distribution, and $S_u$ and $S_m$ for KMM$_{HFS}$ are smaller than those of KMM$_{KTS}$, as revealed in Figure 6. We observe the following.

- For $\nu_a < 1$ min$^2$, $S_u$ and $S_m$ are insensitive to the variance of the inter-arrival time.
- For $\nu_a > 1$ min$^2$, $S_u$ and $S_m$ are increasing functions of $\nu_a$. Specifically, a change in $\nu_a$ significantly affects $S_u$ of KMM and KMM$_{HFS}$.

A large $\nu_a$ means the system traffic becomes more bursty, that is, most inter-arrival times are very small, but a few of them are very large. The inter-arrival times thus diverge in a wide range. If the same period $T_f$ is observed, then the number of UEs joining the service (i.e., $X$) for a large $\nu_a$ is greater than that for a small $\nu_a$, and the effect of increasing $\nu_a$ is similar to the effect of decreasing $\frac{1}{\lambda}$. Therefore, $S_u$ and $S_m$ increase as $\nu_a$ increases. Specifically, if $\nu_a \geq 60$ min$^2$, the $S_u$ of KMM$_{HFS}$ is larger than that of KMM$_{KTS}$.
Effects of $\frac{1}{\lambda}$: Figure 8 plots $S_u$ and $S_m$ as functions of the mean session holding time $\frac{1}{\lambda}$ for KMM, KMMKTS, and KMMHFS, respectively. In these figures, $\frac{1}{\lambda} = 2$ mins; $v_s = 4$ min$^2$; $\lambda = 100$ mins; $v_s = 570$ min$^2$; $\lambda = 60$ mins; $\lambda = 4$; $\Delta_t = 5$ mins; $\Delta_s = 20$ mins. Notably, $X$ is fixed for every $\frac{1}{\lambda}$ and the execution numbers of Procedures P3 and P4 are also fixed. As $\frac{1}{\lambda}$ rises (i.e., the time period that a joined UE remains in the service rises), $Y$ falls and $N$ rises. Since the total traffic to the system is fixed, increasing $\frac{1}{\lambda}$ has the following phenomena.

**Phenomenon 1:** As $\frac{1}{\lambda}$ increases, $Y$ decreases linearly, which implies that the execution number of Procedure P2 of the three schemes decreases.

**Phenomenon 2:** As $\frac{1}{\lambda}$ increases, $N$ increases, which implies that the number of unicast messages for each Procedure P2 of KMM and KMMHFS increases.

**Phenomenon 3:** As $\frac{1}{\eta}$ increases, $N$ increases, which implies that the number of unicast messages for each Procedure P1 of KMM increases.

**Phenomenon 4:** As $\frac{1}{\eta}$ increases, $N$ increases, which implies that the number of unicast messages for each

---

Fig. 6. The effects of $\frac{1}{\lambda}$ on $M_u, S_u, S_m$ and $M_m$ ($\frac{1}{\mu} = 100$ mins; $v_s = 570$ min$^2$; $\frac{1}{\eta} = 60$ mins; $\alpha = 4$; $\Delta_t = 5$ mins; $\Delta_s = 20$ mins).

---

Fig. 7. The effects of $v_u$ on $S_u$ and $S_m$ ($\frac{1}{\lambda} = 2$ mins; $\frac{1}{\lambda} = 100$ mins; $v_s = 570$ min$^2$; $\frac{1}{\lambda} = 60$ mins; $\alpha = 4$; $\Delta_t = 5$ mins; $\Delta_s = 20$ mins).

---

Fig. 8. The effects of $\frac{1}{\lambda}$ on $S_u$ and $S_m$ ($\frac{1}{\lambda} = 2$ mins; $v_s = 4$ min$^2$; $\frac{1}{\lambda} = 100$ mins; $v_s = 570$ min$^2$; $\alpha = 4$; $\Delta_t = 5$ mins; $\Delta_s = 20$ mins).
VI. Conclusion

This study proposed a new scheme for distributing MBMS keys over the UMTS network. Based on the concept of Key Management Mechanism (KMM) proposed by 3GPP, the Key-Tree Scheme (KMMKTS), which works efficiently in wired IP networks, has been modified to fit the mobile environment. Additionally, this study proposed a new scheme, known as a Hash Function Scheme (KMMHFS), in which a hash function is adopted to update the keys S on UEs and the BM-SC. Via the security analysis, we proved the security of KMMHFS.

Simulation experiments have been developed to investigate the performance for KMM, KMMKTS and KMMHFS. We discussed the number of unicast/multicast messages and the number of keys sent on the unicast channels. Simulation results indicate that KMMHFS needs less unicast communication than the other two schemes on Sn and Sm if the traffic is not heavy. When the number of UE arrivals is very large, KMMKTS can reduce the unicast burden (i.e., Sn) but increase the multicast overhead (i.e., Sm). The burst multicast overhead due to the key distribution of KMMKTS may result in that the QoS is not guaranteed in some period with very heavy traffic.

Besides, all UEs listening to the MTCH has to process every received multicast key, where the total computation overhead in all UEs increases. We should notice that the effects of multicast key delivery is more significant than the unicast key delivery. Additionally, KMMKTS requires large memory in UEs. The number of group keys required for a service is lglN, and it increases as more UEs join this service, while only one S is stored for each service in KMM and KMMHFS. Thus, KMMKTS is more complex than the other two schemes. By contrast, KMM and KMMHFS are simple to implement. Compared with KMM, the proposed scheme KMMHFS can reduce overhead in unicast messages, and also provides the same QoS, because KMMHFS does not increase the multicast overhead.

APPENDIX A

The Analytical Model

This section proposes an analytical model to validate the simulation model by calculating the number of occurrences of executions for P1 or P2 during a MBMS session. Consider the timing diagram in Figure 9. Suppose that the elapse time of an MBMS User Service session, Ts, is exponentially distributed with the density function f(t), the mean 1/µ and Laplace transform f*(s) = (µ/µ + s), to validate the simulation model (which is mainly used to conduct our study and is easily applied different distributions for the session times).

During an MBMS User Service session, when an UE joins or leaves the multicast group of this service at his own will, P1 or P2 is triggered. In our analytical model, the UE inter-arrival time for an MBMS User Service session, tu, is assumed to be exponentially distributed with the mean 1/λ. The time period when the joined UE stays in the MBMS User Service session, tu, has the density function fu(t), the mean 1/λ and the Laplace transform fu*(s).

Let X be the number of UEs joining the multicast group conditioning the period Ts. From [22], the probability that X = x during period Ts = t can be expressed as

\[ Pr[X = x|Ts = t] = \frac{(\lambda t)^x}{x!} e^{-\lambda t}. \]

Then, Pr[X = x] (the probability that x UEs join the multicast group during period Ts) can be obtained as follows:

\[ Pr[X = x] = \int_0^\infty Pr[X = x|Ts = t]f(t)dt = \left(\frac{\lambda x}{x!}\right)\int_0^\infty t^x f(t)e^{-\lambda t}dt = \left(\frac{\lambda x}{x!}\right)\left[-(-1)^x \frac{d^x f(s)}{ds^x}\right]|_{s=\lambda} = \frac{\mu x^x}{(\mu + \lambda)^{x+1}}\]

and the expected number of X is given as

\[ E[X] = \sum_{x=1}^\infty x Pr[X = x] = \frac{\lambda}{\mu}. \]

The UEs joining the MBMS User Service within the period Ts may leave the multicast group at any time. Let τ be the time period between the time when the UE joins the multicast group and the time when the session for the service stops. Then from [22], the density function r(τ) for the distribution of τ is expressed

\[ r(\tau) = \mu \int_0^\tau f(t) dt = \mu(1 - F(t))|_{t=\tau} \]

where F(t) is the distribution function of Ts. Since Ts is exponentially distributed, τ and Ts have the same distribution, that is r(τ) = e^{-µτ}.

If the UE leaves the multicast group of the service before the session stops (i.e., tu < τ), the User Service Leave procedure is executed and S is updated. Otherwise (i.e., tu ≥ τ), the current S remains. Let Y be the number of UEs leaving the multicast group during the period Tu. Then, Pr[Y = y|X = x] is the probability that among x joined UEs, y UEs leave (where y ≤ x) the multicast group before the session terminates. Pr[Y = y|X = x] can be computed by counting all possible ways of y UEs leaving the multicast group before the session terminates, which is given by

\[ Pr[Y = y|X = x] = \binom{x}{y}(Pr[t_u < \tau])^y(Pr[t_u \geq \tau])^{x-y}. \]

where Pr[t_u ≥ τ] can be calculated as

\[ Pr[t_u \geq \tau] = \int_{t_u=0}^\infty \int_{t=0}^{t_u} \mu e^{-\mu t}dt f_u(t_u)dt_u = \int_{t_u=0}^{\infty} \int_{t=0}^{t_u} (1 - e^{-\mu t})f_u(t_u)dt_u = 1 - \int_{t_u=0}^{\infty} f_u(t_u)e^{-\mu t}dt_u = 1 - f_u^*(\mu). \]

and Pr[t_u < τ] can be calculated as f_u*(µ).

Thus, (5) is rewritten as

\[ Pr[Y = y|X = x] = \binom{x}{y} f_u^*(\mu)^y [1 - f_u^*(\mu)]^{x-y}. \]
The expected number of \( Y \) is thus expressed as

\[
E[Y] = \sum_{x=1}^{\infty} \sum_{y=1}^{x} \{ y \Pr[Y = y | X = x] \Pr[X = x] \}
\]

\[
= \sum_{x=1}^{\infty} \left\{ \sum_{y=1}^{x} \left( \frac{x}{y} f_u^*(\mu)^y \right) \left[ 1 - f_u^*(\mu)^{x-y} \right] \right\}
\]

\[
\times \left[ \frac{\mu \lambda^x}{(\mu + \lambda)^{x+1}} \right] . \tag{7}
\]

This study takes the Gamma distribution as an example for the distribution of \( t_u \). The Gamma distribution with shape parameter \( \alpha \), the mean \( \frac{1}{\eta} \) and the variance \( \frac{1}{\eta^2} \) has the Laplace transform \( f_u^*(s) \) as

\[
f_u^*(s) = \left( \frac{\alpha \eta}{\alpha \eta + s} \right)^\alpha . \tag{8}
\]

Applying (8) to (7), (7) can be written as

\[
E[Y] = \sum_{x=1}^{\infty} \left\{ \left\{ \sum_{y=1}^{x} \left( \frac{x}{y} \left( \frac{\alpha \eta}{\alpha \eta + \mu} \right)^y \right) \right\} \alpha^y \right\}
\]

\[
\times \left[ 1 - \left( \frac{\alpha \eta}{\alpha \eta + \mu} \right)^\alpha \right] \left[ \frac{\mu \lambda^x}{(\mu + \lambda)^{x+1}} \right] . \tag{9}
\]

Figure 10 plots \( E[Y] \) where \( T_s = 100 \) and \( \frac{1}{\lambda} = 4 \). In this figure, solid curves are the analytical results and the points are the simulation results. The curves and the symbol points show that the analytic analysis is consistent with the simulation results (all errors are within 3\%).

**ACKNOWLEDGMENT**

The authors would like to thank the four anonymous reviewers. Their comments have significantly improved the quality of this paper.

**REFERENCES**


Shin-Ming Cheng received the BS and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2000 and 2007, respectively. Currently, he is a postdoctoral researcher at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. His research interests include mobile networking, wireless communications, and network security.

Wei-Ru Lai received the BSEE and Ph.D. degrees from the Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, R.O.C., in 1991 and 1999, respectively. In 1999, she became an Assistant Professor with the Department of Information Management, Chiming College, Taiwan. She is currently an Assistant Professor with the Communications Engineering Department, Yuan Ze University, Tiao-Yuan, Taiwan, R.O.C. Her research interests include the design and analysis of personal communications services.

Phone Lin (M’02-SM’06) received his BSCSIE degree and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C., in 1996 and 2001, respectively. From August 2001 to July 2004, he was an Assistant Professor in Department of Computer Science and Information Engineering (CSIE), National Taiwan University, R.O.C. Since August 2004, he has been an Associate Professor in Department of CSIE and in Graduate Institute of Networking and Multimedia, National Taiwan University, R.O.C. His current research interests include personal communications services, wireless Internet, and performance modeling. Dr. Lin has published more than twenty international SCI journal papers (most of which are IEEE Transactions and ACM papers). Dr. Lin is an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a Guest Editor for IEEE WIRELESS COMMUNICATIONS special issue on Mobility and Resource Management, and a Guest Editor for ACM/SPRINGER MONET special issue on Wireless Broad Access. He is also an Associate Editorial Member for the WCNC Journal. Dr. Lin has received many research awards. He was elected as the Best Young Researcher, the 3rd IEEE ComSoc Asia-Pacific Young Researcher Award, 2007. He was a recipient of Research Award for Young Researchers from Pan Wen-Yuan Foundation in Taiwan, in 2004, a recipient of K. T. Li Young Researcher Award honored by ACM Taipei Chapter in 2004, a recipient of Wu Ta You Memorial Award of National Science Council (NSC) in Taiwan in 2005, a recipient of Fu Ssu-Nien Award of NTU in 2005 for his research achievements, and a recipient of 2006 Young Electrical Engineering Award, the Chinese Institute of Electrical Engineering. Dr. Lin is listed in Who’s Who in Science and Engineering(R) in 2006. Dr. Lin is a Senior Member, IEEE. P. Lin’s email and website addresses are plin@csie.ntu.edu.tw and http://www.csie.ntu.edu.tw/~plin, respectively.