A Hash Function Scheme for Key Management in UMTS MBMS

Shin-Ming Cheng
Dept. of Computer Science and
Information Engineering
National Taiwan University
Taipei 106, R.O.C.
Email: shimi@pcs.csie.ntu.edu.tw

Wei-Ru Lai
Dept. of Electrical Engineering
Yuan Ze University
Tao-Yuan 320, R.O.C.
Email: wrlai@saturn.yzu.edu.tw

Correspondent Author: Phone Lin
Dept. of Computer Science and
Information Engineering
National Taiwan University
Taipei 106, R.O.C.
Email: plin@csie.ntu.edu.tw

***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. In this paper, we first modify the KTS to be applied in UMTS MBMS. Then we detail HFS. We prove the correctness of HFS. Our study shows that the proposed HFS can reduce both communication and storage overhead without damaging QoS of UMTS MBMS.**

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]. 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. To prevent the non-joined UEs from receiving the MBMS content, 3GPP 33.246 proposed the Key Management Mechanism (KMM) [5], which are described as follows.

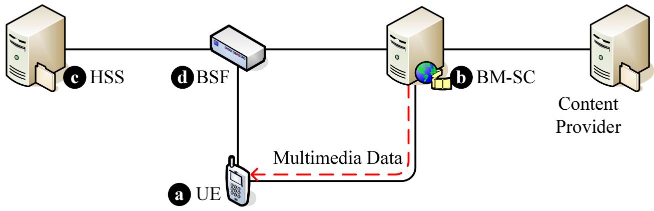

Fig. 1. A simplified UMTS MBMS network architecture

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 multicasted to all joined UEs. A UE uses **T** to decrypt content that it receives. **T** is multicasted from BM-SC to all joined UEs by sending $\mathbf{S}\{\mathbf{T}\}$, 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 `P1` for Event 1), the User Service Leave procedure (denoted as `P2` for Event 2), the MSK Periodic Update procedure (denoted as `P3` for Event 3), and the MTK Periodic Update procedure (denoted as `P4` for Event 4) are exercised at this moment in order to update **T** or **S** [5]. The four procedures are described in detail below.

Figure 2 shows the message flow for Procedure `P1` 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, $\mathbf{S}^{old}$ and $\mathbf{T}^{old}$ are used for the MBMS User Service.

User Service Join Procedure `P1`:

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

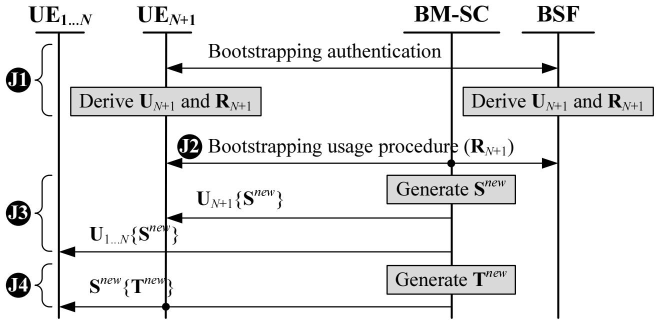
UE$_{1...N}$
UE$_{N+1}$
BM-SC
BSF
J1
Bootstrapping authentication
Derive $\mathbf{U}_{N+1}$ and $\mathbf{R}_{N+1}$
Derive $\mathbf{U}_{N+1}$ and $\mathbf{R}_{N+1}$
J2
Bootstrapping usage procedure ($\mathbf{R}_{N+1}$)
J3
Generate $\mathbf{S}^{new}$
$\mathbf{U}_{N+1}\{\mathbf{S}^{new}\}$
$\mathbf{U}_{1...N}\{\mathbf{S}^{new}\}$
J4
Generate $\mathbf{T}^{new}$
$\mathbf{S}^{new}\{\mathbf{T}^{new}\}$

Fig. 2. Message flow for the User Service Join procedure in KMM

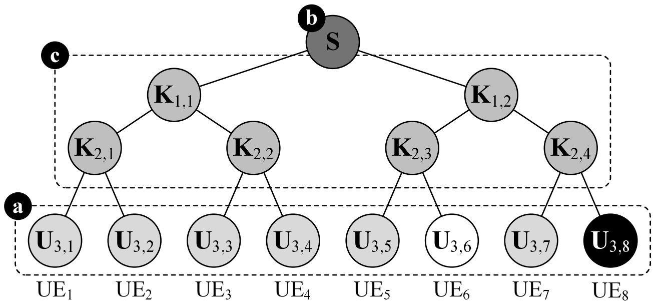

Fig. 3. An example of the key tree in KTS

Step J1. UE$_{N+1}$ performs the bootstapping authentication procedure [3] with BSF to obtain an MBMS Request Key (denoted as $\mathbf{R}_{N+1}$) and an MBMS User Key (denoted as $\mathbf{U}_{N+1}$).

Step J2. UE$_{N+1}$ uses $\mathbf{R}_{N+1}$ as the authentication password when executing the bootstapping usage procedure [3] with BM-SC and BSF.

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

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

The other three procedures are similar to Procedure P1. Procedure P2 consists of three steps, Steps L1–L3, which are the same as Steps J2–J4, respectively. Procedure P3 comprises two steps, Steps S1 and S2, which are the same as Steps J3 and J4, respectively. Procedure P4 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 [6], and **T** is delivered via the MTCH, which is also 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.

Only one group key (that is used for data encryption and unicasted to every member of a multicast group) is defined in IP multicast networks. Previous studies [7], [8], [9], [10] have attempted to reduce the number of uncastings 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. Studies [11], [12] applied KTS to cellular networks in 2004, when the UMTS MBMS was not well defined (i.e., only one group key was considered in these studies).

In this work, KTS is first modified so that it can be applied 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 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 [7], [8]. 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 $H = \lceil \lg N \rceil$. The keys in the tree have the index number (i, j), where $0 \le i < H$ is the layer number, and $1 \le j \le 2^i$ 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$_j$ is assigned the user key $\mathbf{U}_{H,j}$ where $1 \le j \le N$. The content is encrypted by the intermediate key $\mathbf{K}_{i,j}$ before it is multicasted to 2^{H-i} UEs, UE$_{2^{H-i}(j-1)+1}$, UE$_{2^{H-i}(j-1)+2}$, ..., UE$_{2^{H-i}j}$. $\mathbf{U}_{H,j}$ is used to encrypt the key that will be unicasted to UE$_j$. UE$_j$ stores **S**, **T**, $\mathbf{R}_j$, $\mathbf{U}_{H,j}$ and $H-1$ intermediate keys, $\mathbf{K}_{H-1,\lceil \frac{j}{2} \rceil}$, $\mathbf{K}_{H-2,\lceil \frac{j}{2^2} \rceil}$, ..., $\mathbf{K}_{1,\lceil \frac{j}{2^{H-1}} \rceil}$. 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 $\mathbf{K}_{1,1}\{\mathbf{S}^{new}\}$ to UE$_1$, UE$_2$, UE$_3$, UE$_4$ and multicast $\mathbf{K}_{1,2}\{\mathbf{S}^{new}\}$ to UE$_5$, UE$_6$, UE$_7$, 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: Assume that UE$_l$ leaves the multicast group. The group keys (including **S** and $H-1$ KEKs) known by UE$_l$ should be updated so that UE$_l$ cannot decode any future multicast content. $\mathbf{K}^{old}_{H-1,\lceil \frac{l}{2} \rceil}$ is updated to $\mathbf{K}^{new}_{H-1,\lceil \frac{l}{2} \rceil}$; $\mathbf{K}^{old}_{H-2,\lceil \frac{l}{2^2} \rceil}$ is updated to $\mathbf{K}^{new}_{H-2,\lceil \frac{l}{2^2} \rceil}$; ...; $\mathbf{K}^{old}_{1,\lceil \frac{l}{2^{H-1}} \rceil}$ is updated to $\mathbf{K}^{new}_{1,\lceil \frac{l}{2^{H-1}} \rceil}$, and

1930-529X/07/$25.00 © 2007 IEEE

$\mathbf{S}^{old}$ is updated to $\mathbf{S}^{new}$. All newly generated keys should be delivered to all joined UEs that own the old keys. The following actions are taken. The KEK, $\mathbf{K}^{new}_{H-1,\lceil \frac{l}{2} \rceil}$, is unicasted to the other UE that owns $\mathbf{K}^{old}_{H-1,\lceil \frac{l}{2} \rceil}$ (i.e., UE$_{l+1}$ if l is odd, or UE$_{l-1}$ if l is even). $\mathbf{K}^{new}_{H-2,\lceil \frac{l}{2^2} \rceil}$, ..., $\mathbf{K}^{new}_{1,\lceil \frac{l}{2^{H-1}} \rceil}$ and $\mathbf{S}^{new}$ are multicasted to the UEs that own the old keys, and are encrypted with each of their respective children's KEKs. Taking Figure 3 (where there are 7 joined UEs, and UE$_6$ leaves the multicast group) as an example, to deliver $\mathbf{S}^{new}$ to UE$_1$, UE$_2$, ..., UE$_5$ and UE$_7$, BM-SC multicasts $\mathbf{K}^{old}_{1,1}\{\mathbf{S}^{new}\}$ to UE$_1$, UE$_2$, UE$_3$ and UE$_4$, and multicast $\mathbf{K}^{new}_{1,2}\{\mathbf{S}^{new}\}$ to UE$_5$ and UE$_7$. These key deliveries can be performed at Step L2.

Note that the key tree may not be balanced when a UE leaves. As recommended by Moyer et al. [13], 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 [13], 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 $\mathbf{U}$ position in the key tree for UE$'$ by executing the Re-balance algorithm in [13]. Let k be the position number of the found $\mathbf{U}$ position, i.e., UE$'$ is assigned $\mathbf{U}_{H,k}$. To simplify our description, UE$'$ is denoted as UE$_k$ hereafter.

To prevent UE$_k$ from decoding overheard multicast content, $\mathbf{K}^{old}_{H-1,\lceil \frac{k}{2} \rceil}$, $\mathbf{K}^{old}_{H-2,\lceil \frac{k}{2^2} \rceil}$, ..., $\mathbf{K}^{old}_{1,\lceil \frac{k}{2^{H-1}} \rceil}$ and $\mathbf{S}^{old}$ should be updated. The newly generated keys (i.e., $\mathbf{K}^{new}_{H-1,\lceil \frac{k}{2} \rceil}$, $\mathbf{K}^{new}_{H-2,\lceil \frac{k}{2^2} \rceil}$, ..., $\mathbf{K}^{new}_{1,\lceil \frac{k}{2^{H-1}} \rceil}$ and $\mathbf{S}^{new}$) are delivered to all joined UEs that own the old keys, which are encrypted by the old keys. BM-SC then unicasts $\mathbf{U}_{H,k}\{\mathbf{K}^{new}_{H-1,\lceil \frac{p}{2} \rceil}, \mathbf{K}^{new}_{H-2,\lceil \frac{p}{2^2} \rceil}, ..., \mathbf{K}^{new}_{1,\lceil \frac{p}{2^{H-1}} \rceil}, \mathbf{S}^{new}\}$ to UE$_k$. In the example of Figure 3, where UE$_8$ joins the multicast group, BM-SC multicasts $\mathbf{K}^{old}_{1,2}\{\mathbf{K}^{new}_{1,2}\}$ to UE$_5$, UE$_6$ and UE$_7$, and unicasts $\mathbf{U}_{3,8}\{\mathbf{K}^{new}_{2,4}, \mathbf{K}^{new}_{1,2}, \mathbf{S}^{new}\}$ to UE$_8$, in order to deliver $\mathbf{K}^{new}_{1,2}$ to UE$_5$, UE$_6$, UE$_7$ and UE$_8$. These key deliveries are performed at Step J3.

MSK Update Procedure `P3`**:** To update $\mathbf{S}$, the all KEKs and $\mathbf{S}$ in the key tree should be regenerated and unicast to all joined UEs, including $\mathbf{S}$ 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, and 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.

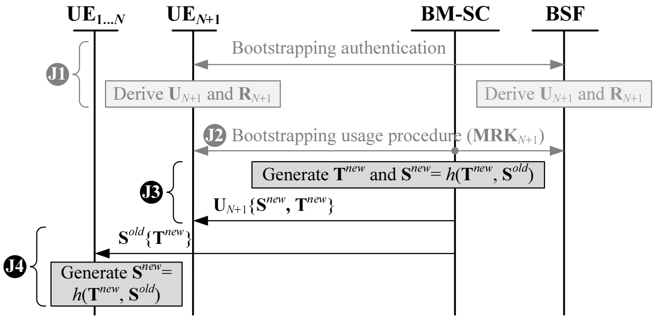
UE$_{1...N}$
UE$_{N+1}$
BM-SC
BSF
J1
Bootstrapping authentication
Derive $\mathbf{U}_{N+1}$ and $\mathbf{R}_{N+1}$
Derive $\mathbf{U}_{N+1}$ and $\mathbf{R}_{N+1}$
J2
Bootstrapping usage procedure ($\mathbf{MRK}_{N+1}$)
J3
Generate $\mathbf{T}^{new}$ and $\mathbf{S}^{new}= h(\mathbf{T}^{new}, \mathbf{S}^{old})$
$\mathbf{U}_{N+1}\{\mathbf{S}^{new}, \mathbf{T}^{new}\}$
$\mathbf{S}^{old}\{\mathbf{T}^{new}\}$
J4
Generate $\mathbf{S}^{new}=$ $h(\mathbf{T}^{new}, \mathbf{S}^{old})$

Fig. 4. Message flow for the User Service Join procedure in HFS

- 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 key tree balanced when UEs join or leave.

The next section proposes the Hash Function Scheme (HFS) for KMM in UMTS MBMS without extra storage space.

III. Hash Function Scheme

A one-way hash function $h(\cdot)$ is a powerful and computationally efficient cryptographic tool [14], 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 $\mathbf{S}$ efficiently. The idea of HFS is that BM-SC requests (through multicast) UEs to generate a new $\mathbf{S}$ by using $h(\cdot)$ instead of unicast $\mathbf{S}$ to all UEs. The HFS exercises as follows. Suppose that the multicast group contains N joined UEs, namely UE$_1$, UE$_2$, ..., UE$_N$, and $\mathbf{S}^{old}$ and $\mathbf{T}^{old}$ are used for the MBMS User Service. To accommodate HFS, Procedures `P1` and `P3` are modified as follows, while the other two procedures are not modified.

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, UE$_{N+1}$, joins the multicast group. If $N=0$ (i.e., UE$_{N+1}$ is the only user in the multicast group), then this procedure is the same as that in KMM of MBMS UMTS. For $N>0$, UE$_{N+1}$ is assigned $\mathbf{U}_{N+1}$ after being successfully authenticated. The BM-SC generates a new $\mathbf{T}$, $\mathbf{T}^{new}$ and a new $\mathbf{S}$ by executing $\mathbf{S}^{new}=h(\mathbf{T}^{new}, \mathbf{S}^{old})$. Then BM-SC unicasts $\mathbf{U}_{N+1}\{\mathbf{S}^{new}, \mathbf{T}^{new}\}$ to UE$_{N+1}$. Then BM-SC multicasts $\mathbf{S}^{old}\{\mathbf{T}^{new}\}$ to the other N joined UEs. The N UEs generate $\mathbf{S}^{new}$ by executing $\mathbf{S}^{new}=h(\mathbf{T}^{new}, \mathbf{S}^{old})$, respectively.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

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

The SHA-1 [15] (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 [14], 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 [14][16], the birthday attack and multicollision attack were proposed to break SHA with less computation overhead, whose details can be found in [14][16]. Wang et al. [17] reduced the complexity of the computation (to find a collision in SHA-1 using collision search attack) to 2^{69}. The computation overhead is still high, i.e., it may cost 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 $\mathbf{S}$ and $\mathbf{T}$ from BM-SC. Thus, HFS is considered robust enough to prevent any birthday and multicollision attacks.

IV. Security Analysis

A secured multicast mechanism should satisfy the group secrecy property [18], which stipulates the following requirements.

- *Nongroup confidentiality*: only the joined UEs can decode the multicast content, i.e., non-joined UEs cannot decode it.
- *Past confidentiality*: a UE joining at time t cannot decode any multicast content before t.
- *Future confidentiality*: 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 [5], in KMM, nongroup confidentiality can be achieved by group keys $\mathbf{S}$ and $\mathbf{T}$, and past and future confidentialities can be achieved via Procedures P1 and P2, respectively. Additionally, in [7], 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 future confidentiality can be achieved. In KMM_{HFS}, we modify Procedures P1 and P3 in KMM. In KMM_{HFS}, Procedure P1 is invoked to update $\mathbf{S}$ and $\mathbf{T}$ when a new UE joins the multicast group at t. Since the UE does not have the old $\mathbf{T}$ and $\mathbf{S}$, he cannot decode any content multicasted before t, and past confidentiality holds in KMM_{HFS}.

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

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

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

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

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

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

Inductive Step: Suppose that the hypothesis holds when $i = k$ (i.e., UE_m cannot get $\mathbf{S}^{(k)}$). 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 $\mathbf{S}^{(k+1)}$ by executing Procedure P1 in KMM_{HFS}, and have

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

where $\mathbf{T}^{(k+1)}$ is delivered by multicast $\mathbf{S}^{(k)}\{\mathbf{T}^{(k+1)}\}$. Since UE_m cannot obtain $\mathbf{S}^{(k)}$, he cannot retrieve $\mathbf{S}^{(k+1)}$.

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

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

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

Thus, the hypothesis holds for all cases.

■

V. Performance Evaluation

We conduct the simulation experiments and mathematical analysis to investigate the performance of KMM_{KTS},

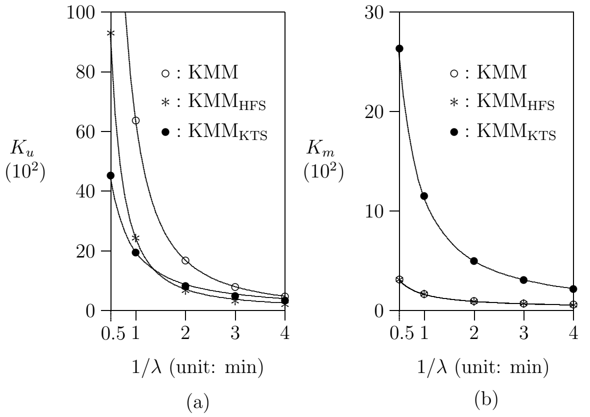

Fig. 5. The effects of $\frac{1}{\lambda}$ on K_u and K_m ($\frac{1}{\mu} = 100$ mins; $v_s = 570$ min^2;$\frac{1}{\eta} = 60$ mins; $\alpha = 4$; $\Delta_t = 5$ mins; $\Delta_s = 20$ mins).

KMM$_{\mathrm{HFS}}$, and KMM, whose details are described in the full paper [19]. This study applies movies as multicast sessions and assume the session service time is Gamma distributed with mean $\frac{1}{\mu}$ and variance v_s. The time when the joined UE stays in a session is assumed to be Gamma distributed with mean $\frac{1}{\eta}$ and the shape parameter α. The UE inter-arrival time to a session is supposed to be exponentially distributed with mean $\frac{1}{\lambda}$. Moreover, **S** and **T** are periodically updated every Δ_s and Δ_t time units, respectively within a session.

Figure 5 plots the experimental results about the performance of total number of keys carried in unicast/multicast messages (denoted as K_u/K_m) for the three schemes, where $\frac{1}{\mu} = 100$ mins, $v_s = 570$ min^2, $\frac{1}{\eta} = 60$ mins, $\alpha = 4, \Delta_t = 5$ mins and $\Delta_s = 20$ mins. Figure 5 (a) indicates when the traffic is small (i.e., $\frac{1}{\lambda} \geq 1.4$ mins), the K_u of KMM$_{\mathrm{HFS}}$ is less than that of KMM$_{\mathrm{KTS}}$. Conversely, when the traffic is large (i.e., $\frac{1}{\lambda} < 1.4$ mins), KMM$_{\mathrm{KTS}}$ requires fewer keys to deliver than KMM$_{\mathrm{HFS}}$. As $\frac{1}{\lambda}$ decreases, more UEs join, stay and then leave the service, and therefore K_u of KMM$_{\mathrm{HFS}}$ increases rapidly. For KMM$_{\mathrm{KTS}}$, K_u increases slowly as $\frac{1}{\lambda}$ decreases.

Figure 5 (b) shows that as $\frac{1}{\lambda}$ decreases, K_m of KMM$_{\mathrm{KTS}}$ increases more rapidly than that of KMM and KMM$_{\mathrm{HFS}}$. Note that K_m are the same for KMM and KMM$_{\mathrm{HFS}}$. Since the number of keys carried in each multicast message at Procedures P1 and P2 is $O(\lg N)$, K_m of KMM$_{\mathrm{KTS}}$ increases rapidly as $\frac{1}{\lambda}$ decreases. All of the multicast messages for KMM$_{\mathrm{KTS}}$ are used for key distribution, and must be transmitted in real-time. Consider a special scenario in which many UEs join or leave simultaneously during a short period. Since P1 and P2 need to transmit many keys and occupy the MTCH, the multimedia content is likely to be compressed, thus degrading QoS. Although KMM$_{\mathrm{KTS}}$ has lower signaling overhead in DCCH, QoS requirement for MTCH may not guaranteed.

Our study indicates that the proposed HFS can reduce communication overhead as UE arrival rate to the session is high. Moreover, KMM$_{\mathrm{HFS}}$ occupies less storage space comparing with KMM$_{\mathrm{KTS}}$.

VI. Conclusion

This study proposes 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 (KMM$_{\mathrm{KTS}}$), which works efficiently in wired IP networks, is modified to fit the mobile environment. Additionally, this study proposes a new scheme, known as a Hash Function Scheme (KMM$_{\mathrm{HFS}}$), in which a hash function is adopted to update **S** on UEs and the BM-SC. The security analysis is presented to prove the security of KMM$_{\mathrm{HFS}}$. From the experimental results, we conclude that the proposed HFS can reduce both communication and storage overhead for UMTS MBMS.

References

[1] 3GPP, “Multimedia Broadcast/Multicast Service; Stage 1 (Release 7),” 3GPP, Tech. Rep. 3G TS 22.146, Mar. 2006.

[2] ——, “Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description (Release 6),” 3GPP, Tech. Rep. 3G TS 23.246, June 2006.

[3] ——, “Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 7),” 3GPP, Tech. Rep. 3G TS 33.220, June 2006.

[4] ——, “Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1 (Release 8),” 3GPP, Tech. Rep. 3G TS 22.246, June 2006.

[5] ——, “3G Security; Security of Multimedia Broadcast/Multicast Service (Release 7),” 3GPP, Tech. Rep. 3G TS 33.246, June 2006.

[6] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, “MIKEY: Multimedia Internet KEYing,” IETF, Tech. Rep. RFC 3830, Aug. 2004.

[7] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, “Multicast Security: Ataxonomy and Some Efficient Constructions,” *IEEE INFOCOM’99*, pp. 708–716, Mar. 1999.

[8] M. J. Moyer, J. R. Rao, and P. Rohatgi, “A Survey of Security Issues in Multicast Communications,” *IEEE Network*, vol. 13, no. 6, pp. 12–23, Nov. 1999.

[9] D. Wallner, E. Harder, and R. Agee, “Key Management for Multicast: Issues and Architectures,” IETF, Tech. Rep. RFC 2627, June 1999.

[10] C. K. Wong, M. Gouda, and S. S. Lam, “Secure Group Communications Using Key Graphs,” *IEEE/ACM Trans. Networking*, vol. 8, no. 1, pp. 16–30, Feb. 2000.

[11] W. Xu, W. Trappe, and S. Paul, “Key Management for 3G MBMS Security,” *IEEE Globecom’04*, pp. 2276–2280, Dec. 2004.

[12] Y. Sun, W. Trappe, and K. J. Liu, “A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks,” *IEEE/ACM Trans. Networking*, vol. 12, no. 4, pp. 653–666, Aug. 2004.

[13] M. J. Moyer, J. R. Rao, and P. Rohatgi, “Maintaining Balanced Key Trees for Secure Multicast,” IETF, Tech. Rep. INTERNET-DRAFT, draft-irtf-smug-key-tree-balance-00.txt, June 1999.

[14] A. Joux, “Collisions for SHA-0,” *Rump session of Crypto’04*, Aug. 2004.

[15] NIST, “FIPS PUB 180-1: Secure Hash Standard,” Apr. 1995.

[16] M. Nandi and D. R. Stinson, “Multicollision Attacks on Some Generalized Sequential Hash Functions,” *IEEE Trans. Inform. Theory*, vol. 53, no. 2, pp. 759–767, Feb. 2007.

[17] X. Wang, Y. L. Yin, and H. Yu, “Finding Collisions in the Full SHA-1,” *Crypto’05*, Aug. 2005.

[18] H. Lu, “A Novel High-Order Tree for Secure Multicast Key Management,” *IEEE Trans. Comput.*, vol. 54, no. 2, pp. 214–224, Feb. 2005.

[19] S.-M. Cheng, W.-R. Lai, P. Lin, and K.-C. Chen, “Key Management for UMTS MBMS,” under preparation.

1930-529X/07/$25.00 © 2007 IEEE