A Study on Key Management for UMTS MBMS Service

鄭欣明
台灣大學　資訊工程學研究所
shimi@pcs.csie.ntu.edu.tw

賴薇如
元智大學　電機工程學研究所
wrlai@saturn.yzu.edu.tw

林風
台灣大學　資訊工程學研究所
plin@csie.ntu.edu.tw

摘要

在網際網路上，群播（multicast）技術已被廣泛的使用來傳送多媒體內容。近年來，行動通訊網路已經和網際網路成功整合。為了將網際網路上的多媒體服務提供給行動裝置存取，第三代行動通訊網路組織（Third Generation Partnership Project; 3GPP）提出了多媒體廣播群播服務（Multimedia Broadcast Multicast Service; MBMS）規範，透過通用行動通訊系統（Universal Mobile Telecommunications System; UMTS）網路將多媒體內容群播給行動裝置。為了讓合法的使用者可以使用 MBMS 服務，3GPP TS 33.246 設計了金鑰管理機制，然而此機制會造成許多額外的信令負荷。本論文針對此問題將傳統的金鑰樹管理機制應用到 MBMS 當中，並將兩種機制詳細分析以及比較。

關鍵詞：金鑰管理（Key Management）、金鑰樹（Key Tree）、多媒體廣播群播服務（MBMS）、通用行動通訊服務（UMTS）

一、簡介

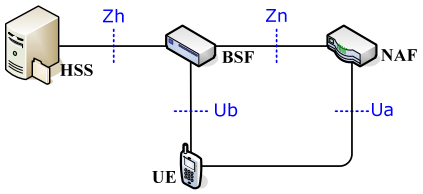
由於 3G 的時代即將到來，各式各樣嶄新的多媒體服務也隨著增加 [20]。而這些服務中絕大部分皆為一個伺服端對多個使用者的方式傳送多媒體資料。在傳統的 IP 網路中，對於此種以群體形式的服務，使用群播的技術可以有效的減少封包的總傳送量。在電信網路當中，第三代行動通訊網路組織（Third Generation Partnership Project; 3GPP）制定了多媒體廣播群播服務（Multimedia Broadcast Multicast Service; MBMS）[2]。

當業者想將此種群播技術在商業上使用時，最重要的就是如何管理使用者存取服務的權限。當使用者加入某個應用服務時，首先廣播群播服務中心（Broadcast Multicast-Service Center; BM-SC）會對這個使用者做認證授權的動作，要是成功之後，BM-SC 則會傳送 MBMS Traffic Key（MTK）給此使用者，系統會將群播的資料用 MTK 作加密，則使用者可以透過 MTK 解開資料。詳細的內容在 3GPP TS 33.246 [7]當中有設計。

當我們學習規範上制定的金鑰管理（Key Management）機制時發現當使用者要離開服務或加入服務或作 MTK 更新時，在 BM-SC 及使用者端之間都需要花費許多的額外信令交換。因此我們嘗試將金鑰樹 [10][14] 的概念套用在 MBMS 的金鑰管理上，這樣的機制的確能有效的減少當使用者進入或離開服務時的信號傳送量，但是其也有些許的缺點：使用者必須花費額外的記憶體儲存更多的金鑰。本篇論文的架構如下：第二章我們介紹 3GPP 中 MBMS 的金鑰管理機制；第三章在 MBMS 的架構當中設計金鑰樹；第四章針對兩種機制做比較；第五章對這篇文章做個結論。

二、MBMS 安全機制

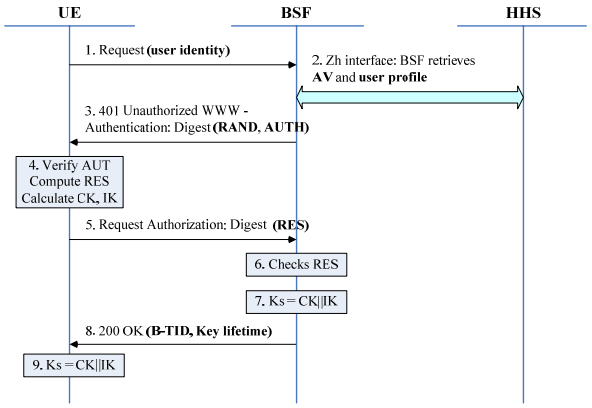
在 UMTS 網路當中，其延續 GSM 網路已經存在一個記錄所有使用者相關資料的資料庫，其名為 Home Subscriber Server（HSS），當使用者透過 User Equipment（UE）存取 UMTS 網路的時候，HSS 必須認證此 UE，確定其是否有權利存取此 UMTS 網路，其中 HSS 將使用 Authentication and Key Agreement（AKA）演算法來計算所需要的金鑰[5]。其認證流程的安全性被廣泛地認定，故 3GPP 在 33.220 [6]將此認證的流程制定成一個更為廣泛的架構稱為 Generic Bootstrapping Architecture（GBA），不論是什麼樣的網路或者是網路中的哪些節點，只要其架構符合 GBA 所定義的，都可以使用 GBA 所提供的認證機制讓 HSS 認證 UE。如圖一所示，此架構有如下的節點與介面：



圖一 GBA 網路架構圖

- 節點 BSF（Bootstrapping Server Function）：BSF 與 UE 之間會執行 AKA 協定以相互認證、得到金鑰。UE 與 NAF 之間可以用此金鑰對於資料作加密。
- 節點 NAF（Network Application Function）：當 bootstrapping 已經完成了之後，UE 和 NAF 之間會執行應用層的安全協定，其認證是基於 BSF 與 UE 雙向認證後得到的金鑰來認證。
- 介面 Ub [3]：提供 UE 與 BSF 之間雙向的認證，允許 UE 根據 3GPP AKA 的架構來 bootstrap 一個金鑰。在 Ub 上面可以執行 HTTP Digest AKA 協定，此協定被定義在 RFC 3310 [11] 並以 3GPP TS 33.102 [5]協定為基礎。

- 介面 Ua [3]：乘載了應用程式協定的封包，其安全性受到金鑰的保護，此金鑰為在 UE 與 BSF 之間的介面 Ub 上執行 HTTP Digest AKA 所得到的結果。



圖二 UMTS Bootstrapping 流程

在 GBA 架構之中有兩個重要的流程，第一個為 Bootstrapping 流程（如圖二所示）細節如下：

步驟 1. 使用者傳送一個 HTTP 的要求給 BSF。訊息內容包含此 UE 的使用者識別碼（即 IMSI）。

步驟 2. BSF 會透過傳送 Authentication Data Request 訊息給 HSS 以取得相關於這個 UE 的認證參數（Authentication Vector; AV）。HSS 透過訊息中 IMSI 得到此 UE 的資料，透過隨機變數 RAND 還有順序數目 SQN 與 UE 都共同擁有的秘密金鑰 K 產生出 AV。其包含隨機變數 RAND、認證訊息 AUTN、金鑰 IK、CK 和用來對照的 XRES。HHS 將數筆 AV 透過 Authentication Data Response 訊息回傳給 BSF。

步驟 3. BSF 會從接收到的 AV 中選擇一個還沒有用過的，將 RAND 與 AUTN 透過 401 Unauthorized WWW Authentication 訊息傳送給 UE。

步驟 4. UE 透過 AUTN 判斷是否能認證此訊息。接著透過 AUTN、RAND 與 K 產生出 SQN、兩組金鑰 IK、CK 與用來對照的 RES。

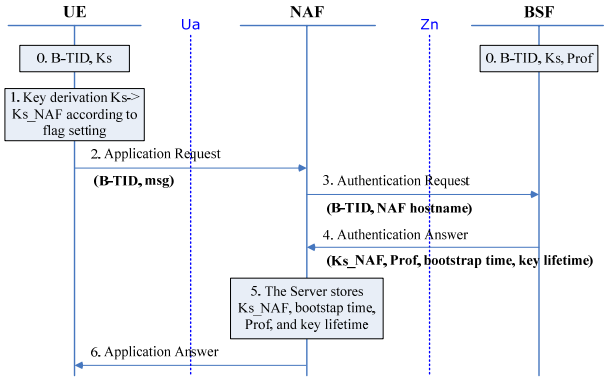
步驟 5. UE 傳送給 BSF 另一個 HTTP 要求，其包含了 RES。

步驟 6. BSF 檢查是否 RES 等於 XRES。

步驟 7. BSF 將 CK 與 IK 結合（concatenate）製造出金鑰 Ks。此時也會產生 Bootstrapping transaction identifier（B-TID）。

步驟 8. BSF 會回傳一個內含 B-TID 的 200 OK 訊息給 UE 表示認證成功。此外，在此 200 OK 訊息裡面，BSF 應該指出金鑰 Ks 的有效時間。

步驟 9. 當 UE 收到 200 OK 訊息，也會用和步驟七同樣的方式產生 Ks。UE 與 BSF 會透過 Ks_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id) 函式產生 Ks_NAF，透過 Ks_NAF 可以保護介面 Ua 的安全性。



圖三 UMTS 使用 Bootstrapped 安全協商流程

第二個流程為使用 Bootstrapped 安全協商流程，如圖三所示，其步驟如下：

步驟 0. 當 UE 執行完圖二的程序後，UE 與 BSF 雙方都已經有 B-TID 與 Ks 了。雙方就會立即開始執行 bootstrapped 安全協商流程。

步驟 1. UE 透過 Ks 和 KDF 產生金鑰 Ks_NAF。

步驟 2. UE 傳送一個 HTTP Request 訊息給 BSF，其中包含 B-TID 的資訊。

步驟 3. NAF 經由介面 Zn 透過 Authentication Request 訊息傳送 B-TID 與其公開的名稱（hostname）給 BSF。

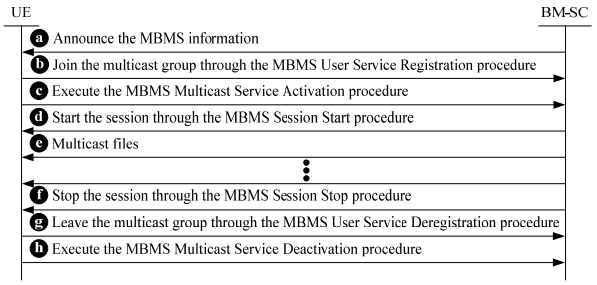
步驟 4. BSF 透過 Ks 與 KDF 產生 Ks_NAF，並將此金鑰、金鑰的有效時間、這次 bootstrapping 的有效時間與此應用程式及 NAF 相關的使用者安全設定透過 Authentication Answer 訊息回傳給 NAF。

步驟 5. NAF 將接收到的訊息儲存。

步驟 6. NAF 回傳 Application Answer 訊息給 UE，則 UE 與 NAF 彼此之間便能夠使用 Ks_NAF 來對資料加解密以達到安全傳送資料的目的。

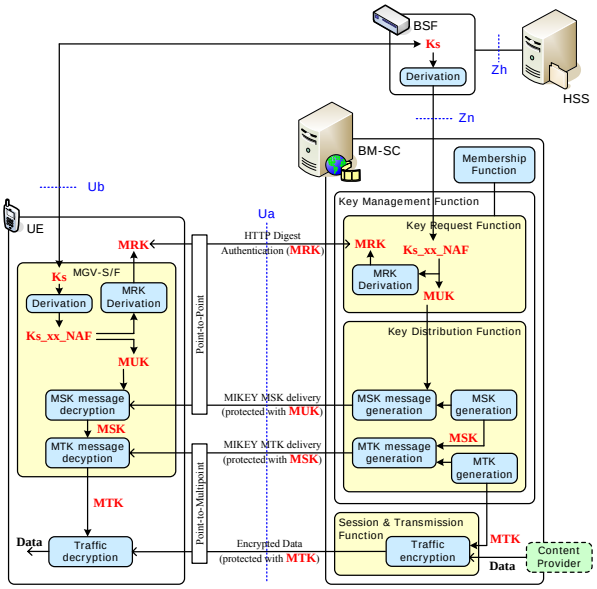
圖四所介紹的 MBMS 服務訊息流程[1][4]。首先 BM-SC 利用 Short Message Cell Broadcast Service 或是 WAP Push 等方式將 MBMS 提供各項服務的相關資訊傳送給 UE（a）。使用者可以了解若要使用此服務，是否需要安全性的考量，若需要的話，在執行 MBMS Multicast Service Activation 程序來加入該 MBMS 群播群組之前（c），會先執行 MBMS User Service Registration 程序與 MBMS User Service 作註冊的動作（b）。此程序是安全的原因為 UE 與 BSF 之間已經執行了上述的 bootstrapping 流程，而 User Service Registration 的實作方式是透過 HTTP-digest 的方式來達成。在此流程當中，BM-SC 的角色即為 GBA 中的 NAF。接著 BM-SC

會執行 MBMS Session Start 程序（d）來 activate 所有需要的載體（bearer）資源，則 BM-SC 即可開始傳送 MBMS 資料給 UE（e）。



圖四 MBMS 服務的訊息流程

當 BM-SC 結束資料的傳送之後，其會執行 MBMS Session Stop 程序（f）來將所有的載體資源釋放。而 UE 可以隨時透過執行 MBMS User Service Deregistration 程序（g）來離開群播群組。最後 UE 執行 MBMS Multicast Service Deactivation 程序（h）來移除網路節點當中所有的繞送資訊。



圖五 MBMS 安全架構

圖五介紹 3GPP 專為 MBMS 群播服務所制定的安全架構 [7]。首先，UE 和 BM-SC 利用 GBA 中的 Bootstrapping 流程產生出一組共同的金鑰（K_S）。而 UE 與 BM-SC 會透過 PDF 將 K_S 產生 MBMS User Key（MUK）及 MBMS Request Key（MRK），每一個使用者裝置都會擁有各自的 MUK 及 MRK。UE 透過圖四中的 MBMS User Service Registration 程序與 BM-SC 註冊某個特定 User Service 時，使用 HTTP Digest [11] 的方式（此訊息中使用的金鑰為 MRK），當 BM-SC 確認了此 UE 之身份之後，會將 UE 所註冊的 User Service 之 MBMS Service Key（MSK）透過此 UE 之 MUK 加密，並傳送給 UE（此時是利用點對點的載體）。而在同一個 User Service 下之 UE 所擁有的 MSK 是相同的，不同的 User Service 會使用不同的 MSK。接著，BM-SC 會利用各 User Service 的 MSK 來加密 MTK，透過群播載體傳送給 UE。最後再利用 MTK 來對真正的群播資料作加密，並在群播載體傳送給 UE。UE 便利用 MTK 來解開群播的資料。

在 UMTS 裡 BM-SC 利用 Multimedia Internet KEYing（MIKEY）協定[12] 跟 UEs 作金鑰交換的工作。其中 MIKEY 訊息中的 General Extension Payload 放置要傳送金鑰的 ID，KEMAC 中放置要傳送的金鑰。UE 收到 MIKEY 訊息之後，由 General Extension Payload 中得知 KEMAK 中放的是什麼金鑰，再利用相對的金鑰（例如 MSK 用 MUK 解密；MTK 用 MSK 解密）來取出收到的金鑰。在 RFC 4563 [13] 中制定了放金鑰的 sub-payload 如表一：

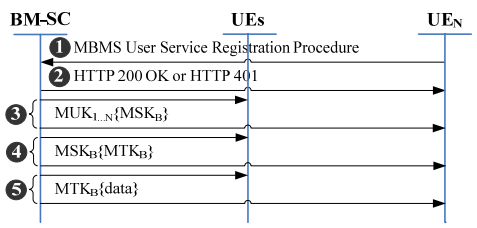
表一 MIKEY 的 sub-payload

Key ID Type	Value	Comment
MBMS Key Domain ID	0	ID of the group key domain
MBMS Service Key ID	1	ID of the group key
MBMS Traffic Key ID	2	ID of the group traffic key

根據 [8][9][15] 的介紹，基於無線網路廣播的特性，一個群播服務必須達到下述四點要求，才能算是一個安全的群播服務：

- 當一個使用者加入某個群組之前，其可以接收到群播服務傳送給其他使用者的封包，此時其無法得到群組金鑰並解開封包。
- 當一個使用者加入到某個群組之後，其得到的群組金鑰不能解開在其加入到群組前已傳送的群播封包（即群組金鑰需要更新）。
- 當一個使用者離開某個群組之後，其無法透過既有的群組金鑰解開此群組中群播的封包（即群組金鑰需要更新）。
- 已離開群組的眾多使用者，並無法依據在離開群組前所得到的知識合力破解出之後的金鑰。

當然，MBMS 群播的安全性也要符合上述四點，接著我們說明在以下需要更換金鑰的情況下，讓如何來確保 MBMS 群播的安全性：



圖六 新使用者加入之流程

- 新使用者加入：如圖六所示，當 UE_N 要加入 MBMS User Service 時，我們必須不讓 UE_N 破解之前的資料，所以在讓 UE_N 加入之前，已在此 User Service 群組中的 UEs 必須更換新的 MSK 和 MTK，再讓 UE_N 取得新的 MSK 和 MTK。其詳細流程討論如下：

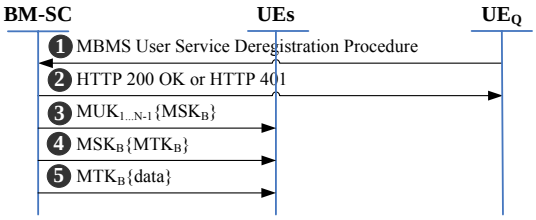
步驟 1. UE_N 利用 HTTP Digest Authentication [11]（包含 user id 與 MRK）向 BM-SC 作 MBMS User Service Registration 程序。

步驟 2. 當 BM-SC 利用相對應的 MRK 認證 UE_N，若成功，則回傳 HTTP 200 OK Authentication-Info，繼續下一步，否則回傳 HTTP 401 WWW-Authentication，則 UE_N 需重新作 MBMS User Service Registration 程序。

步驟 3. 假設 BM-SC 原本是利用 MSK_A 及 MTK_A 來提供服務，BM-SC 利用已在服務中每個 UE 的 MUK 傳送新的 MSK_B 給已在服務中的每個 UE。也利用 UE_N 的 MUK 傳送新的 MSK_B 給 UE_N。

步驟 4. BM-SC 利用新的 MSK_B 加密並群播傳送新的 MTK_B 給所服務中的 UE。

步驟 5. BM-SC 利用新的 MTK_B 繼續提供服務。



圖七 舊使用者離開之流程

- 舊使用者離開：如圖七所示，當有使用者 UE_Q 要離開 MBMS User Service 時，我們必須讓 UE_Q 無法破解離開之後收聽到的資料，所以我們必須更換留在此 User Service 中所有 UEs 的 MSK 和 MTK。其詳細流程討論如下：

步驟 1. UE_N 利用 HTTP Digest Authentication（包含 user id 與 MRK）向 BM-SC 作 MBMS User Service Deregistration 程序。

步驟 2. 當 BM-SC 利用相對應的 MRK 認證 UE_N，若成功，則回傳 HTTP 200 OK Authentication-Info，繼續下一步，否則回傳 HTTP 401 WWW-Authentication，則 UE_N 需重新作 MBMS User Service Deregistration 程序。

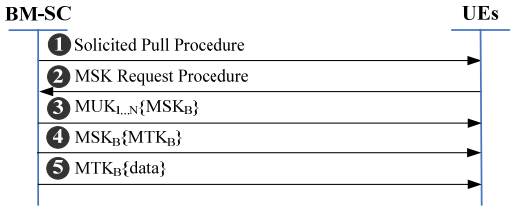
步驟 3. 假設 BM-SC 原本是利用 MSK_A 及 MTK_A 來提供服務，BM-SC 利用已在服務中每個 UE 的 MUK 傳送新的 MSK_B 給已在服務中的每個 UE。

步驟 4. BM-SC 利用新的 MSK_B 加密並群播傳送新的 MTK_B 給所服務中的 UE。

步驟 5. BM-SC 利用新的 MTK_B 繼續提供服務。

- MSK 更新：如圖八所示，為了確保一定的安全性，我們必須按時更換 MSK。

步驟 1. 假設 BM-SC 原本是利用 MSK_A 及 MTK_A 來提供 MBMS User Service，BM-SC 會利用各 UEs 的 MUK 保護傳給 UEs Solicited Pull 的訊息，邀請 UEs 向 BM-SC 要求新的 MSK。



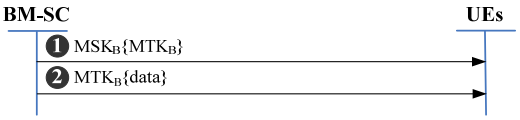
圖八 MSK 更新之流程

步驟 2. UEs 收到 Solicited Pull 的訊息後，向 BM-SC 發送 MSK Request 要求新的 MSK。

步驟 3. BM-SC 利用各 UEs 的 MUK 將新的 MSK_B 傳給在服務中的所有 UEs。

步驟 4. BM-SC 利用新的 MSK_B 加密並群播傳送新的 MTK_B 給所服務中的 UE。

步驟 5. BM-SC 利用新的 MTK_B 繼續提供服務。



圖九 MTK 更新之流程

- MTK 更新：如圖九所示，為了確保一定的安全性，我們必須按時更換 MTK。

步驟 1. BM-SC 利用新的 MSK_B 加密並群播傳送新的 MTK_B 給所服務中的 UE。

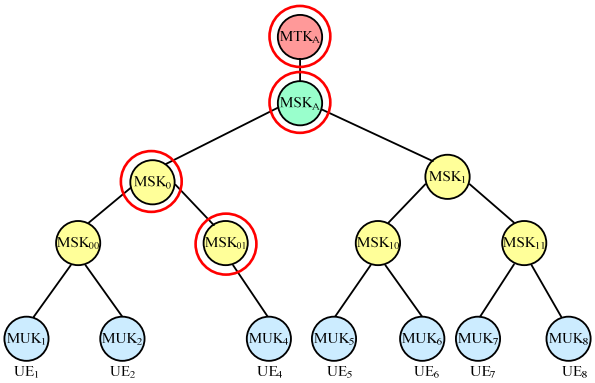
步驟 2. BM-SC 利用新的 MTK_B 繼續提供服務。

綜觀上四種情況，我們得知在 MBMS 中，為了確保其安全性必須常常更換 MSK，而且當 MSK 更換時，所有群組當中的 UE 都必須透過點對點的方式得到新的 MSK。這裡特別注意的原因為規範當中規定 MSK 一定要透過 MUK 加密點對點地傳送，而 MTK 一定要透過 MSK 加密群播地傳送。透過圖六與圖七我們發現，MSK 更換時，其所花費的信令複雜度與此 User Service 的使用者人數成正比，因此若某 MBMS User Service 的 UE 很多，並且 UE 加入離開此服務的次數很頻繁時，更換 MSK 的訊息會對整個網路造成嚴重的負荷。因此我們在下一章中嘗試將已經被證實可以減少信令傳輸量的的金鑰樹管理架構套用在 MBMS 上。

三、在 MBMS 當中應用金鑰樹管理機制

許多研究[10][14][15][16][17][18][19]討論金鑰

樹管理機制，其透過讓群播群組的使用者多儲存一些金鑰，使得金鑰更新時所要耗費的總訊息量大大地減少，因此我們將金鑰樹的架構應用於 MBMS 中。在應用當中，我們期望不更改 MBMS 的金鑰傳送的原則：「MSK 一定要透過 MUK 加密點對點地傳送，而 MTK 一定要透過 MSK 加密群播地傳送」，即表示我們可以實際地將 MIKEY 應用在 MBMS 金鑰樹管理機制當中。如圖十所示，假設某 MBMS User Service 有 7 個 UE 正在使用，則金鑰樹的架構規定每一個 UE 擁有一組金鑰組（key set），例如，UE_4 擁有之金鑰組為 $\{MSK_{01}, MSK_0, MSK_A, MTK_A\}$，其中 MSK_{01} 與 MSK_0 是 UE_4 多擁有的，在金鑰樹架構下我們稱其為 Key Encryption Key（KEK）[16][19]。在本章當中，MSK{MTK}表示透過 MSK 加密 MTK；MSK^*表示是新的 KEK。同上一章，我們考慮下述三種狀況：



圖十 金鑰樹架構

- 使用者 UE_3 加入。

步驟 1. BM-SC 群播 $MSK_{01}\{MSK^*_{01}\}$，由於只有 UE_4 有 MSK_{01}，故其可以解開訊息得到 MSK^*_{01}。

步驟 2. BM-SC 群播 $MSK_0\{MSK^*_0\}$，只有 UE_1、UE_2 與 UE_4 可以解開訊息得到 MSK^*_0。

步驟 3. BM-SC 群播 $MSK_A\{MSK_B\}$，則所有 UE 可以解開訊息得到新的 MSK_B。

步驟 4. 對於 UE_3，BM-SC 利用點對點的方式傳送 $MUK_3\{MSK^*_{01}, MSK^*_0, MSK_B\}$給他。此時所有的 UE（包括 UE_3）都得到了新的 MSK_B。

步驟 5. BM-SC 群播 $MSK_B\{MTK_B\}$給所有 UE。則所有 UE 可用新的 MTK_B 來繼續接收加密的資料。

- 使用者 UE_3 離開。

步驟 1. BM-SC 點對點傳送 $MUK_4\{MSK^*_{01}\}$ 給 UE_4，只有 UE_4 可以解開訊息得到 MSK^*_{01}。

步驟 2. BM-SC 群播 $MSK^*_{01}\{MSK^*_0\}$，UE_4 可以解開得到 MSK^*_0。

步驟 3. BM-SC 群播 $MSK_{00}\{MSK^*_0\}$，則 UE_1 與 UE_2 可以解開得到 MSK^*_0。

步驟 4. BM-SC 透過群播 $MSK^*_0\{MSK_B\}$讓 UE_1、UE_2 與 UE_4 解開訊息得到新的 MSK_B。

步驟 5. BM-SC 透過群播 $MSK_1\{MSK_B\}$讓 UE_5 到 UE_8 得到新的 MSK_B。此時所有的 UE 都得到了 MSK_B。

步驟 6. BM-SC 群播 $MSK_B\{MTK_B\}$給所有 UE。則所有 UE 可用新的 MTK_B 來繼續接收加密的資料。

- MSK 更新。為了維持 MBMS 金鑰傳送的規則，此步驟會稍微複雜。

步驟 1. BM-SC 點對點傳送 $MUK_1\{MSK^*_{00}\}$ 給 UE_1、傳送 $MUK_2\{MSK^*_{00}\}$ 給 UE_2、傳送 $MUK_3\{MSK^*_{01}\}$給 UE_3、傳送 $MUK_4\{MSK^*_{01}\}$給 UE_4、傳送 $MUK_5\{MSK^*_{10}\}$ 給 UE_5、傳送 $MUK_6\{MSK^*_{10}\}$給 UE_6、傳送 $MUK_7\{MSK^*_{11}\}$給 UE_7、傳送 $MUK_8\{MSK^*_{11}\}$給 UE_8。

步驟 2. BM-SC 群播 $MSK^*_{00}\{MSK^*_0\}$讓 UE_1與 UE_2 可解開得到新的 MSK^*_0。

步驟 3. BM-SC 群播 $MSK^*_{01}\{MSK^*_0\}$讓 UE_3與 UE_4 可解開得到新的 MSK^*_0。

步驟 4. BM-SC 群播 $MSK^*_{10}\{MSK^*_1\}$讓 UE_5與 UE_6 可解開得到新的 MSK^*_1。

步驟 5. BM-SC 群播 $MSK^*_{11}\{MSK^*_1\}$ 讓 UE_7 與 $UE8_4$ 可解開得到新的 MSK^*_1。

步驟 6. BM-SC 群播 $MSK^*_0\{MSK_B\}$讓 UE_1 到 UE_4 可解開得到新的 MSK_B。

步驟 7. BM-SC 群播 $MSK^*_1\{MSK_B\}$讓 UE_5 到 UE_8 可解開得到新的 MSK_B。此時所有的 UE（包括 UE_3）都得到了 MSK_B。

步驟 8. BM-SC 群播 $MSK_B\{MTK_B\}$給所有 UE。則所有 UE 可用新的 MTK_B 來繼續接收加密的資料。

由上述三點我們可以得知為了維護金鑰樹的架構，在更新金鑰的時候，需要花費額外的資訊傳送 KEK，即便如此，當一個群播群組的人數為 n 時，金鑰樹架構已經被證明金鑰更新（不論有人離開會進來）的信令複雜度為 $\Theta(\log n)$[10][14]。然而金鑰更新時期複雜度 $\Theta(n)$，其原因為為了配合 MSK 一定要透過 MUK 加密點對點地傳送的原則（即步驟 1），但是讀者應該有發現 KEK 的傳送也是透過 MSK，這表示我們要稍微更改 MIKEY 的協定已達到如此的能力，如此一來，我們可以將金鑰更新的複雜度降下到 $\Theta(\log n)$。

然而，金鑰樹的架構仍有缺點，即 UE 必須儲存許多的 KEK，當金鑰樹的深度越大時，UE 必須儲存越多的 KEK，對於目前的記憶體有限的 UE，這樣的機制會造成額外的負擔。

四、比較

在本章我們比較上述兩種機制的優缺點，如表二所示，我們可以得知 3GPP 的金鑰管理機制會造成大量的信令負荷，而金鑰樹管理機制雖然可以有效地減少信令負荷，但是為了維持每個 UE 上的金鑰集，其信令負荷複雜度會有一個底線（lower bound），即 Θ (log n)，且這些金鑰集會對 UE 上的記憶體造成額外的負荷。

表二 兩個機制的比較

	3GPP	Key Tree
Communication Complexity per join	Θ (n)	Θ (log n)
Communication Complexity per leave	Θ (n)	Θ (log n)
Communication Complexity per update	Θ (n)	Θ (log n)
Memory Complexity in UE	Θ (1)	Θ (log n)

五、結論

透過本論文讀者可以詳細瞭解 MBMS 的架構、運作流程，本論文也介紹了 MBMS 的安全性機制，在安全性的考量下，如何兼顧安全性又不造成控制信令的浪費為其重要的考量。由於在傳統的 MBMS 金鑰管理機制中，當金鑰更新時需要花費許多額外的訊息，因此我們期望金鑰樹應用在 MBMS 當中以減少信令量。在設計整個 MBMS 的金鑰樹機制時，我們期望不違背 MBMS 安全機制的一些設定，即表示我們可以實際地將 MIKEY 應用在 MBMS 金鑰樹管理機制當中。。透過詳細的分析比較，我們可以得知金鑰樹管理機制在安全性上、信令傳送複雜度上得到較好的效能，然而規範所設計的機制在 UE 記憶體使用複雜度上取得較好的效能。

參考文獻

[1] 3GPP. Multimedia Broadcast/Multicast Service (MBMS); MBMS user services. TS 22.246, 3GPP, June 2006.

[2] 3GPP. Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description. TS 23.246, 3GPP, June 2006.

[3] 3GPP. Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details TS 24.109, 3GPP, June 2006.

[4] 3GPP. Multimedia Broadcast Multicast Service (MBMS); Protocols and codecs. TS 26.346, 3GPP, June 2006.

[5] 3GPP 3G Security; Security architecture (Release 6). TS 33.102, 3GPP, December 2004

[6] 3GPP 3G Security; Generic Authentication Architecture (GAA); Generic bootstrapping architecture. TS 33.220, 3GPP, June 2006

[7] 3GPP 3G Security; Security of Multimedia Broadcast Multicast Service. TS 33.246, 3GPP, June 2006

[8] Harney, H. and Muckenhirn, C., Group Key Management Protocol (GKMP) Specification, RFC 2093, IETF, July 1999

[9] Harney, H. and Muckenhirn, C., Group Key Management Protocol (GKMP) Architecture, RFC 2094, IETF, July 1999

[10] Wallner, D., Harder, E., and Agee, R. Key Management for Multicast: Issues and Architectures, RFC 2627, IETF, June 1999.

[11] Niemi A., Arkko, J., and Torvinen, V., Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA), RFC 3310, IETF, September 2002

[12] Arkko, J., Garrara, E., Lindholm, F., Naslund, M., and Norrman, K., MIKEY: Multimedia Internet KEYing, RFC 3830, IETF, August 2004

[13] Carrara, E., Lehtovirta, V., and Norrman, K., The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY), RFC 4563, IETF, June 2006

[14] Canetti, R., Garay, J., Itkis, G., Miccianancio, D., Naor, M., and Pinkas, B., Multicast security: A Taxonomy and Some Efficient Constructions, *IEEE INFOCOM '99*, 2:708-716, March 1999

[15] Moyer, M. J., Rao, J. R., and Rohatgi, P., A Survey of Security Issues in Multicast Communications, *IEEE Network*, 13(6):12–23, November–December 1999

[16] Wong, C., Gouda, M., and Lam, S., Secure Group Communications Using Key Graphs, *IEEE/ACM Transactions Networking*, 8(1):16-30, February 2000

[17] Judge, P. and Ammar, M., Gothic: A Group Access Control Architecture for Secure Multicast and Anycast, *IEEE INFOCOM '99*, 1547-1566, June 2002

[18] Wenyuan X., Trappe, W., and Paul, S., Key Management for 3G MBMS Security, *IEEE GLOBECOM '04*. 4: 2276-2280, 29 November–3 December 2004

[19] Sun, Y., Trappe, W., and Liu, K. J., A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks, *IEEE/ACM Transactions on Networking*. 12(4): 653-666, August. 2004

[20] Etoh, M. and Yoshimura, T., Wireless Video Applications in 3G and Beyond, *IEEE Wireless Communications*, 12(4): 66-72, August 2005