動態群組之可量化安全金鑰管理

吳士慶
國立雲林科技大學
電子與資訊工程研究所
雲林縣斗六市大學路三段 123 號
shihchingwu@yahoo.com.tw

伍麗樵
國立雲林科技大學
電子與資訊工程研究所
雲林縣斗六市大學路三段 123 號
wuulc@el.yuntech.edu.tw

摘要

近年來以群組為基礎的網路應用程式愈來愈風行，為了有效利用網路資源，需採用群播來傳送群組資料，然而目前的群播機制並不提供任何的安全服務，容易受到有心人士的破壞，必需利用密碼學及金匙管理的機制來保護在網路上傳送的群播資料。

本文提出一應用於安全群組通訊的階層式金匙管理機制，能提供群組通訊時傳送資料的私密性、完整性及來源認證性。利用分層管理的概念將整個群組畫分成路由區域與成員區域，達到金匙管理的可量化性；而延伸 Diffie-Hellman Key Exchange 方法來產生各個區域的輔助金匙，使得群組成員所需持有的輔助金匙數量最小化，即成員只需擁有一把輔助金匙，而管理員擁有兩把輔助金匙。此外，本文的金匙管理機制並不要求在群組通訊開始前就事先決定群組的成員數目，即本文的方法適用於成員關係經常變動的動態群組。

關鍵詞：金鑰管理、安全群播、Diffie-Hellman 金匙交換、群組金匙、輔助金匙

一、研究動機與目的

近年來由於網際網路的蓬勃發展，人們利用網路來進行溝通已成為一種新趨勢，許多新興的網路通訊應用軟體相應而生，如使用視訊會議(Video Conference)讓遠方的人員亦可參與會議；使用遠距教學(Interactive Distance Education)的方式讓修課的學生不再侷限於一定要在教室聽課，透過網路一樣能在家裡或是其它地方聽課…等諸如此類的應用。上述的應用如果沿用傳統點對點(peer to peer)傳輸機制來傳送群組資料，不僅對發送者的系統造成負擔，而且還會耗用大量的網路資源來傳送重複的資料封包給多個接收者。利用群播(multicast)[1]機制可以有效的利用網路資源，避免不必要的資源浪費。

然而早期的群播機制並不提供任何的安全服務，有心人士可以很容易地在網路上取得群組通訊內容，並進行竄改與假冒等的破壞行為。我們必需利用密碼學及金匙管理的機制來保護在網路上傳送的群播資料，即建構一具可量化性及有效率的安全群播機制。

所謂安全群播[2]就是在群組成員溝通時，能提供資料的私密性(privacy)、完整性(integrity)及可認證性(authentication)的機制，也就是說必須要滿足下列幾點的特性：

1. 非群組的成員不能夠得知群組成員之間資料傳送的內容。
2. 成員間資料的傳送必須提供有來源端認證的機制。
3. 一個新加入的成員，不能夠得知他在加入群組之前，群組成員間資料傳送的內容。
4. 一個已經離開群組的成員，不能得知其離開群組後，群組成員間資料傳送的內容。

要達到上述安全群播的特性，必需在群組成員之間建立一把共享的加解密金匙，這把加解密金匙稱為群組金匙(Group Key)，所有成員利用群組金匙來加解密群組通訊內容，非群組成員則因為無法取得群組金匙而無法得知

群組通訊的內容為何。

一般安全群播需要建立一群組金匙的管理及更新機制，安全地將群組金匙提供給所有的群組成員，而在群組成員關係發生變動時(新成員加入或是舊有成員離開群組通訊)，則必須進行群組金匙的更新(Re-keying)。通常，群組成員在參與群組通訊的過程中，除了擁有一把群組金匙用來加解密群組資料之外，還會擁有一把或是數把輔助金匙(Auxiliary Key)用來幫助群組金匙的更新。

本文的研究目的在提出一應用於安全群組通訊的階層式金匙管理機制，能提供群組通訊時傳送資料的私密性，完整性及來源認證性，我們稱為動態群組之可量化安全金鑰管理(Scalable Secure Key Management for Dynamic Multicast Group，簡稱為 SSKM)。我們所提出的 SSKM 具備下列特性：

1. 利用分層管理的概念將整個群組畫分成路由器區域與成員區域，達到金匙管理的可量化性(Scalability)。
2. 延伸 Diffie-Hellman Key Exchange 方法[4]來產生各個區域的輔助金匙，使得群組成員所需持有的輔助金匙數量最小化，即成員只需擁有一把輔助金匙，而管理員擁有兩把輔助金匙。
3. 金匙管理機制並不要求在群組通訊開始前就事先決定群組的成員數目，適用於成員關係經常變動的動態群組(Dynamic Group)。
4. 利用群播的方式更新群組金匙及傳送計算輔助金匙所需的資料，使得更新金匙所需的訊息量大為減少。由運算能力較強大的管理員負責產生輔助金匙的大部分計算負擔，群組成員則只需執行一次指數運算便可求出其所需的輔助金匙。

二、相關研究

(一)Diffie-Hellman Key Exchange

這是一種很有名的公開金匙交換演算法，最早是在 1976 年被提出[3]，目的是讓兩個使用者(訊息發送者與訊息接收者)可以安全地各自產生一把相同的金匙。表 2.1 簡單說明其演算法的條件及運算流程。

表 2.1Diffie-Hellman Key Exchange

若α是質數q的一個 primitive root，則 $\alpha \bmod q, \alpha^2 \bmod q, \cdots, \alpha^{q-1} \bmod q$ 皆不同，且其值所成的集合為 $\{1, 2, \cdots, q-1\}$。
Global Public Elements q Prime number α $\alpha < q$ and α is a primitive root of q
User A Key Generation Select private X_A $X_A < q$ Calculate public Y_A $Y_A = \alpha^{X_A} \bmod q$
User B Key Generation Select private X_B $X_B < q$ Calculate public Y_B $Y_B = \alpha^{X_A} \bmod q$
Generation of Secret Key by User A $K = (Y_B)^{X_A} \bmod q$
Generation of Secret Key by User B $K = (Y_A)^{X_B} \bmod q$

使用者A與使用者B最後都可以分別獨立且安全地求出其共同的秘密金匙K，其他人即使在網路上竊聽到Y_A及Y_B亦無法求出K。這是因為即使已知$y = \alpha^x \bmod q$、α及q的值，要求出相對應的 x 值，必需花費$e^{((\ln q)^{1/3} \ln(\ln q))^{2/3}}$的時間[22]。因此當我們所選取的質數值非常大的時候，要有效率地求出相對應的 x 值則是不可行的。

(二)金鑰建立協定

如何在開放性的網路上舉行一安全群組通訊，所有的群組成員必須透過一協定來建立一把共享的加解密金匙，用來加解密群組通訊的內容，此種類型的協定稱為金鑰建立協定(key establishment protocol)。透過此種協定，只有群組成員才能得到這把群組金匙，其他非群組成員則無法取得這把群組金匙，群組成員使用這一把金匙將群組通訊內容加密及解密，彼此才能安全地互相溝通。根據群組金匙決定的方式與分配的過程之不同，我們又可以將金鑰建立協定區分成兩種類型：金鑰協調協定(Key Agreement Protocols)[3-6]，另一種則是金匙分配協定(Key Distribution Protocols)[7-17]，接下來我們詳細說明這兩者的不同之處。

(1)金鑰協調協定

金鑰協調協定的特徵在於其群組金匙是由群組成員在加入群組通訊時選擇一隨機變數，之後再經由交換資訊後共同決定，並非由某一群組成員或管理者單獨決定，每個成員對於群組金匙有相同的決定權力。近年來有許多金鑰協調協定的論文[3-6]被提出。由於每次新的群組金匙的建立都需要所有的群組成員參與共同合作，它具有以下之缺點：

1. 群組成員可能散佈於網際網路的各個地方，所有成員必須遵循協定的正確順序，交換彼此資訊，以順利產生群組金匙。然而在分散式的環境之下，這並不是一件容易做到的工作。
2. 由於每個成員所擁有的系統設備並不相同，是否每個成員都有足夠能力能順利執行協定中所有程序。
3. 其協定不具可量化性，當群組成員大量增加，其成員間的通訊量會大幅增加，並不適合在大型且群組成員關係經常變動的群組通訊中使用。

(2)金匙分配協定

金匙分配協定的特徵在於其群組金匙是由群組通訊的發起者或由群組管理員產生，透過事先與群組成員間建立的安全溝通通道(secure communication channel)，安全地將群組金匙分配給所有成員；而在成員關係有所變動時，亦必須將新的群組金匙分配給所有的參與者，以維持安全群播的特性。通常，群組成員在參與安全群組通訊的過程中，除了擁有一把群組金匙用來加解密群組資料之外，還會擁有輔助金匙用來幫助群組金匙的更新。

輔助金匙架構的選擇將影響到群組的量化問題，在[7]中針對已經提出的三種輔助金匙架構：**N root/leaf pairwise keys approach**[18-19]、**Complementary variable approach** 及 **Hierarchical tree approach**[11]所需輔助金匙的數量及更新群組金匙所需的訊息數量有詳細的討論。

C.K.Wong[11]將階層的觀念應用到群組金匙與輔助金匙的管理上，雖然需要較多數量的輔助金匙，但是在應用到動態的群組成員關係時，能提供較有效率的更新群組金匙的機制，達到可量化的要求。在 G.Caronni[12]及I.Chang[8]中，除了利用階層的輔助金匙架構，並配合不同的金匙配置方式，減少在C.K.Wong[11]中所需要的輔助金匙數目。在這三篇研究都是利用階層式的輔助金匙架構來有效率的更新群組金匙，他們的缺點是：僅是單一成員關係的變動(例如成員加入或離去)，卻影響到全部群組成員必需進行群組金匙及多數輔助金匙的更新動作。

L.C.Wuu, H.C.Chen[13]亦利用分層管理的概念將整個群播群組分成路由器區域與成員區域，並分別以霍夫曼樹及完全二元樹的架構來建立其對應的輔助金匙分散樹，其利用群組的分層管理及樹狀的輔助金匙架構，使其群播金匙的管理具備可量化的特性。但是其缺點則是，所有成員必需事先經過註冊成為註冊成員，爾後再經由加入群組的動作而成為群組成員，群播群組則因為其樹狀的輔助金匙架構而有人數上限的限制，且其中每個人擁有的輔助金匙亦根據其群組成員人數上限與其樹狀的輔助金匙架構而不同，群組成員人數上限越高，每個成員加入群組通訊時所擁有的輔助金匙數也越多。

三、基本架構

(一)二階層群組管理架構

在 SSKM 中，為了有效利用網際網路架構的特性，我們將整個群組區分成兩個階層來管理：第一層稱為路由器區域(Router Domain)，由具備有群播能力的路由器所組成；第二層包含多個成員區域(Member Domain)，每個成員區域由一個具備群播能力的路由器及至少一個群組成員(Group Member)所組成。

在每一個區域中都會有一個管理員(Manager)，負責管理該區域的成員關係及協助輔助金匙的產生。群組發起者(Group Initiator)所在區域的路由器則是群組管理員(Group Manager)，負責管理路由器區域內所有的子群組管理員、產生群組金匙及協助“管理員輔助金匙”的產生。而其他的路由器則是子群組管理員(Subgroup Manager)，負責管理該成員區域內的群組成員關係及協助“成員輔

助金匙”的產生。圖 3.1 說明我們所使用的二階層群組管理的架構。

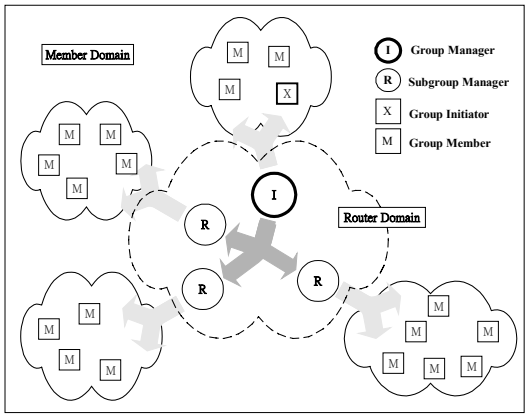

圖 3.1 二階層群組管理架構

SSKM 利用分層管理的概念，將整個群組成員的管理負擔分散到各個子群組管理員上，讓群組管理員負責管理所有的子群組管理員，而各個子群組管理員負責管理其管轄區域內的群組成員，利用這樣階層化的概念，將使得動態群組的管理擁有可量化的特性。

接下來解釋在本文中使用到的符號，茲列表說明如下。本文假設在每一個區域內都存在一個認證中心，每個認證中心彼此信任對方，而且每個使用者或是路由器在加入群組通訊之前都擁有經由任證中心認證過的公開/私人金匙。

表 3.1 符號解釋

符號	說明
G	群組名稱
I	群組管理員
R	子群組管理員
M	群組成員
n	群組成員總數；n_R 則表示在 R 子群組的成員個數
$GrpKey$	群組金匙
$AKey$	管理員輔助金匙：一把由所有管理員共享的輔助金匙
$MAKey$	成員輔助金匙：每個子群組各有一把，由該子群組管理員與該管轄範圍內的群組成員共享。
q	質數
α	$\alpha < q$, α is a primitive root of q
s_R	管理員 R 隨機選取的私密分享值，用來計算管理員輔助金匙
m_A	使用者 A 隨機選取的私密分享值，用來計算成員輔助金匙
K_A, K_A^{-1}	使用者 A 的公開/私人金鑰
$[Data]K_A^{-1}$	Data 以金匙 K_A^{-1} 加密
$\{Data\}K_A^{-1}$	使用者 A 對 Data 做數位簽章
$\langle Data\rangle K_A^{-1}$	$\langle Data\rangle K_A^{-1}$ = Data, $\{Data\}K_A^{-1}$
$\rightarrow$	單播
$\mapsto$	群播
$\Rightarrow$	廣播

(二)SSKM 基本運作概論

SSKM 的基本運作大致上可以區分成兩大部分來說明，分別是註冊程序(Registration Procedure) 與金匙更新策略 (Re-keying Strategy)。

(1)註冊程序

註冊程序發生在群組通訊開始形成時，由群組發起者發出一個新群組通訊的公告，所有對該群組有興趣的使用者都可以執行註冊程序加入群組通訊。由群組管理員自行產生群組金匙，並且延伸 Diffie-Hellman Key Exchange 方法來產生“管理員輔助金匙”及“成員輔助金匙”。群組管理員用“管理員輔助金匙”加密群組金匙後群播給每個子群組管理員，由子群組管理員解密後取得群組金匙，再用其所在區域的“成員輔助金匙”加密後群播傳送給群組成員，當群組成員得到群組金匙後即可加入該群組，進行安全的群組通訊。

● 管理員輔助金匙

群組管理員取得所有子群組管理員的私密分享值，再利用自己的私密分享值與之做運算後產生“管理員輔助金匙”，群組管理員隨後將子群組管理員計算“管理員輔助金匙”所需的資料群播傳送給每個子群組管理員；子群組管理員取得所需的資料，再用自己的私密分享值做運算產生同一把“管理員輔助金匙”。

我們以一個簡單的例子來說明，圖 3.2 是在群組通訊的路由器區域內“管理員輔助金匙”產生過程的簡單示意圖，其中 I 為群組發起者所在區域的路由器，也就是群組管理員，I 選定一私密分享值為 s_I。R_2、R_3 和 R_4 則是所有的子群組管理員，其選取的私密分享值分別為 s_2、s_3 及 s_4，並且利用 I 的公開金匙加密後傳送給 I，只有 I 才能解密取得 s_2、s_3 及 s_4。

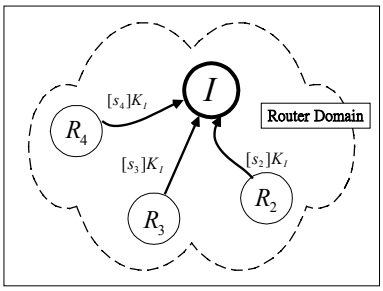

圖 3.2 管理員輔助金匙產生過程示意圖

則 I 可以求出其“管理員輔助金匙” $AKey = \alpha_I^{s_I s_2 s_3 s_4} \bmod q_I$，並群播 $\{ \alpha_I^{s_I s_3 s_4} \bmod q_I$，$\alpha_I^{s_I s_2 s_4} \bmod q_I$，$\alpha_I^{s_I s_2 s_3} \bmod q_I, [GrpKey]AKey \}$給所有子群組管理員，隨後由子群組管理員自行求出 $AKey$。

如子群組管理員 R_2 利用 s_2 及 $\alpha_I^{s_I s_3 s_4} \bmod q_I$ 做計算後求得 $AKey = (\alpha_I^{s_I s_3 s_4})^{s_2} \bmod q_I$，$R_3$ 則用 s_3 及 $\alpha_I^{s_I s_2 s_4} \bmod q_I$ 做計算後求得 $AKey = (\alpha_I^{s_I s_2 s_4})^{s_3} \bmod q_I$。求得 $AKey$ 後將 $[GrpKey]AKey$ 解密以取得群組金匙，然後再由各個子群組管理員分配群組金匙給群組成員。

● 成員輔助金匙

子群組管理員取得其管轄區域所有群組成員的私密分享值，再利用自己的私密分享值與之做運算後產生“成員輔助金匙”，子群組管理員隨後群播計算“成員輔助金匙”所需的資料給該區域群組成員，群組成員取得所需的計算值，再用自己的私密分享值做運算後產生同一把“成員輔助金匙”。

(2)金匙更新策略

金匙更新策略因成員加入或是成員離開情形的不同而有所不同，整個系統必需重新產生新的群組金匙及相關的輔助金匙，並且利用輔助金匙更新群組金匙，以下簡單說明更新過程。

成員加入指的是使用者欲加入目前已經存在網路上的群組通訊，使用者發出加入要求(join request)後，由群組管理員發起更新群組金匙，群組管理員利用目前的群組金匙加密新群組金匙後群播傳送給所有群組成員，完成更新群組金匙的動作；然而新群組成員加入的成員區域因為有新成員加入，所以其所在區域的“成員輔助金匙”必需重新產生，新的“成員輔助金匙”必需結合新成員的私密分享值計算而成。

成員離開指的是群組成員欲離開目前的群組通訊，成員發出離開要求(leave request)後，由群組管理員發起更新群組金匙，由於離開的成員擁有目前的群組金匙，所以群組管理員改利用“管理員輔助金匙”加密新的群組金匙後群播傳送給所有子群組管理員，子群組管理員再利用“成員輔助金匙”加密新群組金匙後群播傳送給群組成員，完成更新群組金匙的動作。

特別注意的是，群組成員離開的成員區域因為有成員離開，所以其所在區域的“成員輔助金匙”必需重新產生，由子群組管理員協助產生，新的“成員輔助金匙”不再利用離開成員的私密分享值計算而成，但該子群組管理員須使用不同的私密分享值，以避免產生相同的“成員輔助金匙”。

當群組成員在進行安全的群組通訊時，除了可以利用群組金匙的加密來達到保密的功能外，亦提供來源認證的功能。即群組成員在發送訊息時，加上對該訊息的簽章，訊息的接受者在收到訊息之後，便可以進行身分驗證。

四、註冊程序

註冊程序詳細的流程列於表 4.1，我們直接以範例來說明 SSKM 的註冊程序是如何運作，解釋輔助金匙如何產生，如何利用“管理員輔助金匙”與“成員輔助金匙”去協助群組金匙的分配。

在圖 4.1 中，我們假設沒有編號的使用者 M 不想加入群組通訊 G，而有編號的使用者為想加入群組通訊 G 的使用者，如 M_2、M_3、…、M_{10}，並且假設其選取的私密值分別為 m_2、m_3、…、m_{10}。

步驟(1)：群組發起者 X 發送一個“群組初始化”($GrpInit$)訊息給它的區域路由器 I，表示即將舉行一群組通訊 G，I 成為群組通訊 G 的群組管理員，且解密 $GrpInit$ 訊息得知 M_X 的私密分享值 m_X。

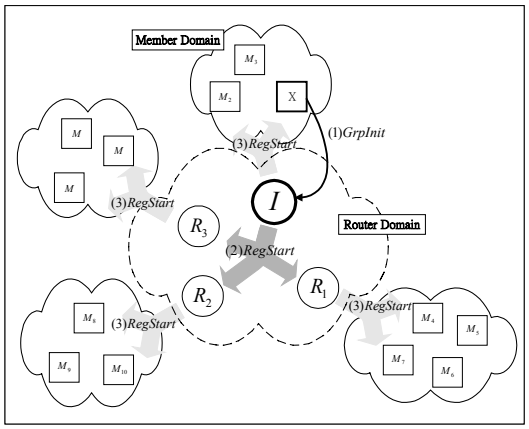

圖 4.1 註冊程序(1)

步驟(2)：I 廣播一“註冊開始”(*RegStart*)訊息給網 R_1、R_2 及 R_3。

步驟(3)：每個路由器(I、R_1、R_2 及 R_3)隨後亦廣播一“註冊開始”(*RegStart*)訊息給在區域網路上的所有使用者，通知所有使用者可以開始註冊加入群組通訊。

步驟(4)：有意加入 G 的使用者傳送一個“註冊”訊息給它自己的區域路由器，該訊息包含使用者隨機選取的私密分享值。M_2 和 M_3 各自傳送“註冊”訊息給 I，I 解密“註冊”訊息得知 M_2 和 M_3 的私密分享值 m_2 和 m_3，且 I 成為他們的子群組管理員。同理，R_1 成為 M_4、M_5、M_6 和 M_7 的子群組管理員，且知道私密分享值 m_4、m_5、m_6 和 m_7。R_2 成為 M_8、M_9 和 M_{10} 的子群組管理員，且知道私密分享值 m_8、m_9 和 m_{10}。

觀察圖 4.1 和圖 4.2，R_3 因為其所管轄區域並沒有使用者想要加入群組通訊 G，R_3 不執行步驟(5)及其以後的程序，所以並沒有出現在圖 4.2，R_1 與 R_2 則繼續繼續下面的步驟。

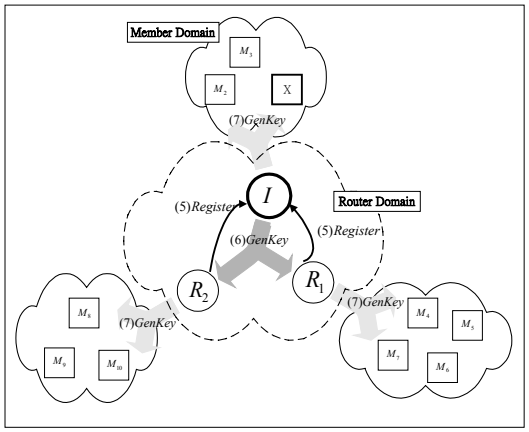

圖 4.2 註冊程序(2)

步驟(5)：子群組管理員 R_1 和 R_2 各自傳送一個“註冊”的訊息給群組管理員 I，I 解密得知 R_1(R_2)的私密分享值 s_1(s_2)。在此一步驟 I 取得所有子群組管理員的私密分享值，在註冊時間 $RegTime_I$ 截止之後產生群組金是 $GrpKey$ 及“管理員輔助金是” $AKey = (\alpha_I^{s_1 s_2})^{s_I} \bmod q_I$。

$$I \mapsto R_1, R_2 : GenKey(< q_I, \alpha_I^{s_I s_2} \bmod q_I, \alpha_I^{s_I s_1} \bmod q_I, [GrpKey]AKey, T_I > K_I^{-1})$$

步驟(6)：I 群播 *GenKey* 訊息給所有的子群組管理員，該訊息內容包含計算“管理員輔助金是”所需之資料及 $[GrpKey]AKey$。

R_1 收到 I 的 *GenKey* 訊息後，取得 q_I 與 $\alpha_I^{s_I s_2} \bmod q_I$，再用 s_1 便可以計算出“管理員輔助金是” $AKey = (\alpha_I^{s_I s_2})^{s_1} \bmod q_I$，並用以解密 $[GrpKey]AKey$ 取得群組金是。同理，R_2 也可以算出“管理員輔助金是” $AKey = (\alpha_I^{s_1 s_1})^{s_2} \bmod q_I$，並解密取得群組金是。

$$I \mapsto M_X, M_2, M_3 : GenKey(< q_I, \alpha_I^{m_I m_2 m_3}, \alpha_I^{m_I m_X m_3}, \alpha_I^{m_I m_X m_2}, [GrpKey]MAKey_I, T_I > K_I^{-1})$$

步驟(7)：我們先看到 I 成員區域的情形。I 群播 *GenKey* 訊息給管轄區域內的群組成員，該訊息內容包含 q_I、計算“成員輔助金是” $MAKey_I$ 所需之資料及 $[GrpKey]MAKey_I$。

M_2 收到 I 的 *GenKey* 訊息後，取得 q_I 與 $\alpha_I^{m_I m_X m_3} \bmod q_I$，再用 m_2 便可以計算出 $MAKey_I = (\alpha_I^{m_I m_X m_3})^{m_2} \bmod q_I$，並用以解密 $[GrpKey]MAKey_I$ 取得群組金是。I 成員區域的其它群組成員亦可依同樣方法求出 $MAKey_I$，然後解密 $[GrpKey]MAKey_I$ 取得群組金是。

同理，其他子群組管理員群播 *GenKey* 訊息給其管轄區域內的群組成員，該訊息內容包含計算“成員輔助金是”所需之資料及用“成員輔助金是”加密群組金是後的值。群組成員則用自己的私密值求出“成員輔助金是”後，再解密取得群組金是。

五、金匙更新策略

當群組成員關係發生變動，像新成員加入或是舊有成員離開群組通訊，就必須執行金匙更新程序，以維持安全群播的特性。在 SSKM 中，將由群組管理員來發起群組金匙的更新程序。接下來則利用範例來詳細說明在成員加入及成員離開時，SSKM 更新策略的詳細流程(表 5.1 至表 5.4)。

(一)成員加入

當有新成員要加入群組通訊時，如果不進行群組金匙的更新，只是單純地將正用於群組通訊的群組金匙分配給新加入的成員，則該名新成員就有可能利用得到的群組金匙去解密取得他尚未加入群組之前的通訊資料，為了維持安全群播的特性，群組管理員必須進行群組金匙更新的程序。

接下來我們以兩種情況來討論新成員加入群組通訊時的金匙更新程序：

1. 成員區域內第一位成員加入，即該成員區域之子群組管理員尚未加入群組通訊(表 5.1)。
2. 成員區域內新成員加入，即該成員區域之子群組管理員已經加入群組通訊(表 5.2)。

礙於篇幅關係，僅介紹成員加入第一種情形。

Case1：子群組管理員尚未加入群組通訊

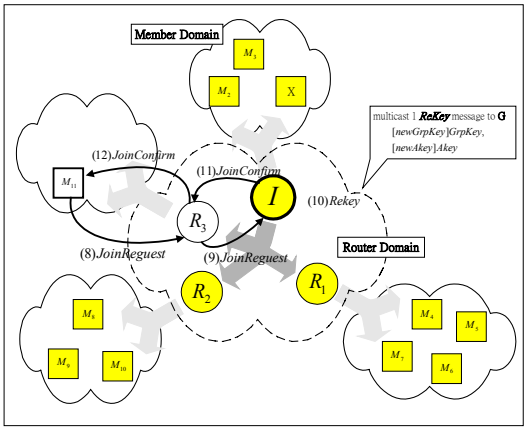

圖 5.1 子群組管理員尚未加入群組通訊

$$M_{11} \rightarrow R_3 : JoinRequest([< G, m_{11}, T_{M_{11}} > K_{M_{11}}^{-1}]K_{R_3})$$

步驟(8)：如圖 5.1 所示，在成員區域 R_3 中，M_{11} 為第一個提出加入群組通訊要求的使用者。使用者 M_{11} 向路由器 R_3 發出“加入要求”(*JoinRequest*)訊息，要求加入群組通訊，並且利用 R_3 的公開金匙將該訊息加密，只有 R_3 才能將該訊息解密取得內容，該訊息內容包含 M_{11} 的私密分享值 m_{11} 以及 Timestamp $T_{M_{11}}$。

$$R_3 \rightarrow I : JoinRequest([< G, s_3, T_{R_3} > K_{R_3}^{-1}]K_I)$$

步驟(9)：此時由於路由器 R_3 尚未加入群組通訊 G，R_3 亦必須向群組管理員 I 發出“加入要求”(*JoinRequest*)訊息，要求加入群組通訊，並且利用 I 的公開金匙將該訊息加密，只有 I 才能將該訊息解密取得內容，該訊息內容包含 R_3 的私密分享值 s_3 以及 Timestamp T_{R_3}。

群組管理員 I 驗證完 R_3 提出的“加入要求”訊息，解密取得 s_3，並選取新的私密值 s_I' 隨即開始產生新的群組金匙 *newGrpKey* 及計算新的“管理員輔助金匙” *newAKey*，新的管理員輔助金匙則是利用 s_I'、s_1、s_2 及 s_3 計算求得 $newAKey = \alpha_I^{s_I' s_1 s_2 s_3} \bmod q_I$。

$$I \mapsto G : Rekey(< [newGrpKey]GrpKey, [newAKey]AKey, T_I > K_I^{-1})$$

步驟(10)：I 向群組 G 群播一個“更新金匙”(*Rekey*)訊息，訊息的內容包含了用目前的群組金匙加密新的群組金匙 $[newGrpKey]GrpKey$、用目前的“管理員輔助金匙”加密新的“管理員輔助金匙” $[newAKey]AKey$ 及 Time Stamp T_I。

除了新加入的群組成員之外，其他所有的群組成員能經由該訊息取得新的群組金匙，而除了新加入的子群組管理員之外，其他所有子群組管理員亦都能經由該訊息取得新的群組金匙及新的“管理員輔助金匙”。

$$I \rightarrow R_3 : JoinConfirm(< q_I, \alpha_I^{s_I' s_1 s_2 \ldots s_u}, [newGrpKey]newAKey, T_I > K_I^{-1})$$

步驟(11)：I 向新加入的子群組管理員回應一個”加入確認”(*JoinConfirm*)訊息，訊息的內容包含計算新的“管理員輔助金

匙”所需的資料、$[newGrpKey]newAKey$ 及 Time Stamp(T_I)。

R_3 經由該訊息取得 q_I 及 $\alpha_I^{s_I s_1 s_2}$，再加上自己所擁有的 s_3，求出新的“管理員輔助金匙” $newAKey = (\alpha_I^{s_I s_1 s_2})^{s_R} \bmod q_I$，並將 $[newGrpKey]newAKey$ 解密取得新的群組金匙。

子群組管理員 R_3 在要求加入群組通訊時，除了選取私密分享值 s_3 供計算“管理員輔助金匙”使用外，亦選取私密分享值 m_{R_3}、α_3 和 q_3 供計算 R_3 之“成員輔助金匙” $MAKey_3$。加上在步驟(8)取得的 m_{11}，求出其“成員輔助金匙” $MAKey_3 = \alpha_3^{m_{R_3} m_{11}} \bmod q_3$。

$$R_3 \rightarrow M_{11} : JoinConfirm(< q_3, \alpha^{m_{R_3}}, [newGrpKey]MAKey_3, T_{R_3} > K_{R_3}^{-1}$$

步驟(12)：R_3 接著向新成員 M_{11} 回應一個”加入認可”訊息，訊息內容包含計算“成員輔助金匙”所需的資料及 $[newGrpKey]MAKey_3$。

M_{11} 經由該訊息取得 q_3、$\alpha_3^{m_{R_3}} \bmod q_3$，再加上自己所擁有的 m_{11}，求出其所在成員區域之“成員輔助金匙” $MAKey_3 = (\alpha_3^{m_{R_3}})^{m_{11}} \bmod q_3$，並可將 $[newGrpKey]MAKey_3$ 解密取得新群組金匙。M_{11} 隨後即可開始進行安全的群組通訊。

(二)成員離開

群組成員會離開群組通訊的主要因素為成員本身要離開群組通訊或者由群組管理員(子群組管理員)強制該成員離開群組通訊。如果不進行群組金匙的更新，則該名離去的成員就有可能利用目前的群組金匙去解密取得他離開群組之後的通訊資料，甚至取得新的群組金匙。為了維持安全群播的特性，為了讓離去的成員不能取得新的群組金匙，我們必須進行群組金匙的更新，即利用輔助金匙來幫助群組金匙的更新。

接下來我們以兩種情況來討論成員離開群組通訊時的金匙更新程序：

1. 成員區域內某一成員離開群組通訊，該成員區域之子群組管理員不需要離開群組通訊(表 5.3)。
2. 成員區域內最後一位成員離開，該成員區域之子群組管理員必需離開群組通訊(表 5.4)。

礙於篇幅關係，僅介紹成員離開第二種情形。

Case2：子群組管理員必須離開群組通訊

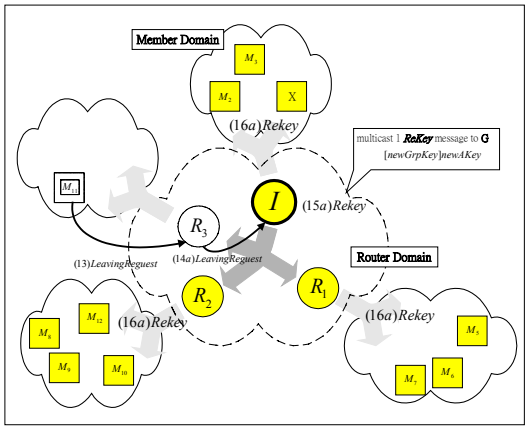

圖 5.2 子群組管理員必須離開群組通訊

$$M_{11} \rightarrow R_3 : LeavingRequest(< G, T_{M_{11}} > K_{M_{11}}^{-1})$$

步驟(13)：如圖 5.2 所示，R_3 成員區域內的最後一位成員 M_{11} 欲離開群組通訊，M_{11} 向 R_3 發出”離開要求”訊息，要求離開群組通訊。

$$R_3 \rightarrow I : LeavingRequest(< G, T_{R_3} > K_{R_3}^{-1})$$

步驟(14a)：由於 R_3 成員區域內的最後一位群組成員 M_{11} 要求離開群組通訊，R_3 亦必需離開群組通訊，向群組管理員 I 發出”離開要求”($LeavingRequest$)訊息，要求離開群組通訊。

群組管理員 I 驗證完子群組管理員 R_3 提出的”離開要求”訊息後，隨即開始產生新的群組金匙 $newGrpKey$ 及計算新的“管理員輔助金匙” $newAKey$，$newGrpKey$ 由群組管理員自行決定，而新的“管理員輔助金匙”則是由群組管理員選取新的私密分享值 s_I'，加上其他子群組管理員之私密分享值計算而成，即 $newAKey = \alpha_I^{s_I' s_1 s_2} \bmod q_I$。

$$I \mapsto R_1, R_2 : Rekey(<\{\alpha^{s_I s_1}, \alpha^{s_I s_2}\}, [newGrpKey]newAKey, T_I > K_I^{-1})$$

步驟(15a)：I 向路由器區域群播一個”更新金匙”(*Rekey*)訊息，訊息的內容包含 $\{\alpha^{s_I s_2}, \alpha^{s_I s_1}\}$、用新的“管理員輔助金匙”加密新的群組金匙 $[newGrpKey]newAKey$。

所有子群組管理員都能利用自己的私密分享值及經由上述之群播訊息之內容，計算產生新的“管理員輔助金匙”，並可解密 $[newGrpKey]newAKey$ 取得新的群組金匙。

For R_1, R_2 *without member leaving*

$$R_1 \mapsto \forall M_{R_1} : Rekey([newGrpKey]MAKey_1, T_{R_1} > K_{R_1}^{-1})$$
$$R_2 \mapsto \forall M_{R_2} : Rekey([newGrpKey]MAKey_2, T_{R_2} > K_{R_2}^{-1})$$

步驟(16a)：此一步驟由沒有成員離開之子群組管理員執行，群播一個”更新金匙”(*Rekey*)訊息給管轄區域之成員，訊息的內容用“成員輔助金匙”加密新的群組金匙 $[newGrpKey]MAKey$、。

R_1 成員區域所有群組成員便可用“成員輔助金匙” $MAKey_1$ 解密 $[newGrpKey]MAKey_1$ 取得新的群組金匙。

六、成果比較

金匙管理機制的設計將直接影響到更新群組金匙時所需要的訊息數及每個群組成員參與群組通訊時所需維護的輔助金匙數目。接下來，我們將對 SSKM 和[9]、[10]、[11]、[12]、[8]、[13]等六篇研究的輔助金匙數量及金匙更新訊息數作一比較。其中[9]、[10]為群組階層化管理的相關研究，而[11]、[12]、[8]、[13]是輔助金匙的相關研究。

在步驟(6)，我們已經假設 SSKM 將群組階層化後有一個群組管理員及 u 個子群組管理員，所以總共產生 u +1 個成員區域(Member $Domain_0$~Member $Domain_u$)，Member $Domain_0$ 為群組管理員所管理的成員區域。Member $Domain_i$ 裡面的群組成員個數為 n_i，群組成員個數為 $n = \sum_{i=0}^{u} n_i$。

(1)金匙數目比較(表 6.1)

SSKM 將群組分成一個路由器區域及 u +1 個成員區域。在路由器區域中只需一把輔助金匙，也就是“管理員輔助金匙” $AKey$；而在每個成員區域所需要的輔助金匙也是一把，也就是該成員區域的“成員輔助金匙” $MAKey_i$，所以總共需要 u +2 把輔助金匙。

(2)金匙更新訊息數目比較(表 6.2)

表 6.2 為新成員加入或是舊有成員離開群組通訊時所需的金匙更新訊息數目的比較。我們假設在加密金匙更新訊息時，對一固定長度金匙的加密動作視為一個金匙更新訊息，例如[K1]K3 是利用 K3 來更新 K1，算一個金匙更新訊息，而[K1,K2]K3，是利用 K3 來更新 K1 及 K2，由於加密的資料量加倍所以算二個金匙更新訊息。

此外，管理員在更新輔助金匙時需要計算成員需要的輔助金匙計算值，然後群播傳送給成員，由成員利用自己的私密值作運算產生所需的輔助金匙。這對管理員來說也是一項蠻重大的負擔，所以我們在比較時亦將管理員計算所需要的計算值個數加入更新訊息數做比較。

七、結論

SSKM 利用分層管理的概念將整個群組畫分成路由區域與成員區域，由群組管理員管理子群組管理員，子群組管理員管理群組成員，達到群組管理可量化性。並延伸利用 Diffie-Hellman Key Exchange 方法來產生各個區域的輔助金匙，使得每個成員參與群組通訊時所需的金匙數固定為二，一為用來加密群組資料的群組金匙，一則是用來輔助更新群組金匙用的“成員輔助金匙”。

此外，由運算能力較強大的管理員協助更新“成員輔助金匙”，負責大部分的計算負擔，讓群組成員則只需一次指數運算便可求出其所需的“成員輔助金匙”。管理員並利用群播將計算“成員輔助金匙”所需的資料傳送給該區域的所有成員，結果使得更新金匙時所需的訊息量大為減少。

八、參考文獻

1. J.A.Kreibich, “The MBONE: the Internet's other backbone”, available from http://www.acm.org/crossroads/xrds2-1/mbone.html.

2. M.J.Moyer, J.R.Rao, P.Rohatgi, “A survey of security issues in multicast communications”, IEEE Network, vol. 13, no. 6, Nov. 1999, pp.12-23.

3. W.Diffie ,M.Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, vol. 22, no. 6, Nov.1976, pp.644–654.

4. I.Ingemarsson, et al, “A conference key distribution system”, IEEE Trans, 1982, IT-28, pp.714-720.

5. M.Steiner, et al., “Diffie-Hellman key distribution extended to group communication”, Proceedings of the 3rd ACM conference on Computer and communications security, 1996, pp.31 – 37.

6. M.Steiner, et al., “New multiparty authentication services and key agreement protocols”, IEEE Journal on Selected Areas in Communications, vol. 18, no.4, April. 2000, pp.628 –639.

7. D.Wallner, et al., “Key Management for Multicast: Issues and Architectures”, RFC 2627.

8. I.Chang, et al., “Key management for secure Internet multicast using Boolean function minimization techniques”, INFOCOM '99, pp.689 -698.

9. S.Mittra, “Iolus: A Framework for Scalable Secure Multicasting”, Proceedings of ACM SIGCOMM’97, Cannes, France, 1997, pp.277-288.

10. T.Hardjono, B.Cain, “Secure and Scalable Inter-Domain Group Key Management for N-to-N Multicast”, 1998 International Conference on Parallel and Distributed Systems, 1998, pp.478-485.

11. C.K.Wong, et al., ”Secure Group Communications Using Key Groups”, IEEE/ACM Transactions on Networking, vol. 8, no. 1, Feb. 2000, pp.16–30.

12. G.Garonni, et al., “Efficient security for large and dynamic multicast groups”, Seventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative, 1998, pp.376-383.

13. L.C.Wuu, H.C.Chen, “Two Level Multicast Key Management”, Journal of Internet Technology, 2000, pp53-58.

14. M.Waldvogel, et al., “The VersaKey framework: versatile group key management”, IEEE Journal on Selected Areas in Communications, vol. 17, no. 9, Sept. 1999, pp.1614-1631.

15. L.R.Dondeti, S.Mukherjee, A.Samal, “DISEC: a distributed framework for scalable secure many-to-many communication”, Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on, 2000, pp.693 –698.

16. M.Refik, P.Alain, “Scalable multicast security in dynamic groups”, Proceedings of the 6th ACM conference on Computer and communications security, 1999, pp.101-112.

17. B.Bob, F.Ian, “Nark: receiver-based multicast non-repudiation and key management”, Proceedings of the first ACM conference on Electronic commerce, 1999, pp.22-30.

18. H.Harney, “Group Key Management Protocol (GKMP) Specification”, RFC 2093.

19. H.Harney, “Group Key Management Protocol (GKMP) Architecture”, RFC 2094.

20. R.Boivie, N.Feldman, C.Metz, “Small group multicast: a new solution for multicasting on the Internet”, IEEE Internet Computing, vol.4, no.3 , May-June 2000 , pp.75–79.

21. U.M.Maurer, “Secret key agreement by public discussion from common information”, IEEE Transactions on Information Theory, vol. 39,no. 3, May 1993, pp.733-742.

22. S.William, Cryptography and Network Security: Principles and Practice, second education, Prentice-Hall, Inc.,1999.

表 4.1 註冊程序

$$X \to I : GrpInit([< G, Info, m_X, T_X > K_X^{-1}]K_I) \quad (1)$$
$$I \Rightarrow all\ routers : RegStart(< G, Info, RegTime_I, T_I > K_I^{-1}) \quad (2)$$
$$R \Rightarrow local\ hosts : RegStart(< G, Info, RegTime_R, T_R > K_R^{-1}) \quad (3)$$
$$M \to R : Register([< G, m_M, T_M > K_M^{-1}]K_R) \quad (4)$$
$$R \to I : Register([< G, s_R, T_R > K_R^{-1}]K_I) \quad (5)$$
$$I \mapsto \forall R : GenKey(< q_I, \{\alpha_I^{s_I \Pi(s_p | p \in [1,u] \wedge p \neq j)} \mid j \in [1,u]\}, [GrpKey]AKey, T_I > K_I^{-1}) \quad (6)$$
$$For\ each\ subgroup\ manager\ R \quad (7)$$
$$R \mapsto \forall M_R : GenKey(< q_R, \{\alpha_R^{m_R \Pi(m_p | p \in [1,n_R] \wedge p \neq j)} \mid j \in [1,n_R]\}, [GrpKey]MAKey_R, T_R > K_R^{-1})$$

表 5.1 新成員加入-子群組管理員尚未加入群組通訊

$$M \to R : JoinRequest([< G, m_M, T_M > K_M^{-1}]K_R) \quad (8)$$
$$R \to I : JoinRequest([< G, s_R, T_R > K_R^{-1}]K_I) \quad (9)$$
$$I \mapsto G : Rekey(< [newGrpKey]GrpKey, [newAKey]AKey, T_I > K_I^{-1}) \quad (10)$$
$$I \to R : JoinConfirm(< q_I, \alpha_I^{s_I' s_1 s_2 \ldots s_u}, [newGrpKey]newAKey, T_I > K_I^{-1}) \quad (11)$$
$$R \to M : JoinConfirm(< q_R, \alpha^{m_R}, [newGrpKey]MAKey_R, T_R > K_R^{-1}) \quad (12)$$

表 5.2 新成員加入-子群組管理員已經加入群組通訊

$$M \to R : JoinRequest([< G, m_M, T_M > K_M^{-1}]K_R) \quad (8)$$
$$R \to I : RekeyRequest([< G, "join", T_R > K_R^{-1}]K_I) \quad (9a)$$
$$I \mapsto G : Rekey(< [newGrpKey]GrpKey, T_I > K_I^{-1}) \quad (10a)$$
$$R \mapsto \forall M_R : Rekey(< [newMAKey_R]MAKey_R, T_R > K_R^{-1}) \quad (11a)$$
$$R \to M : Joinconfirm(< q_R, \alpha_R^{m_R' m_1 m_2 \ldots m_{nR}}, [newGrpKey]newMAKey_R, T_R > K_R^{-1}) \quad (12a)$$

表 5.3 成員離開-子群組管理員不需離開群組通訊

$$M \to R : LeavingRequest(< G, T_M > K_M^{-1}) \quad (13)$$
$$R \to I : RekeyRequest(< G, "leaving", T_R > K_R^{-1}) \quad (14)$$
$$I \mapsto \forall R : Rekey(< [newGrpKey]AKey, T_I > K_I^{-1}) \quad (15)$$
$$For\ R_x\ without\ member\ leaving \quad (16)$$
$$R_x \mapsto \forall M_{R_x} : Rekey([newGrpKey]MAKey_{R_x}, T_{R_x} > K_{R_x}^{-1})$$
$$For\ R\ with\ \ member\ leaving \quad (17)$$
$$R \mapsto \forall M_R : Rekey(\{\alpha^{\frac{m_R'}{m_M}\Pi(m_p | p \in [1,n_R] \wedge p \neq j)} \mid j \in [1,n_R]\}, [newGrpKey]newMAKey_R, T_R > K_R^{-1})$$

表 5.4 成員離開-子群組管理員必需離開群組通訊

$$M \to R : LeavingRequest(< G, T_M > K_M^{-1}) \quad (13)$$
$$R \to I : LeavingRequest(< G, T_R > K_R^{-1}) \quad (14a)$$
$$I \mapsto \forall R_x : Rekey(< \{\alpha^{\frac{s_I'}{s_R}\Pi(s_p | p \in [1,u] \wedge p \neq j)} \mid j \in [1,u]\}, [newGrpKey]newAKey, T_I > K_I^{-1}) \quad (15a)$$
$$For\ R_x\ without\ member\ leaving \quad (16a)$$
$$R_x \mapsto \forall M_{R_x} : Rekey([newGrpKey]MAKey_{R_x}, T_{R_x} > K_{R_x}^{-1})$$

表 6.1 金匙數目比較

	[9]	[10]	[11] with degree d	[12]	[8]	[13]	SSKM
Group Key	0	1	1	1	1	1	1
Subgroup Key	$u+1$	$u+1$	0	0	0	$u+2$	0
Auxiliary Key	$N+u$	$N+u$	$\frac{d}{d-1}(N-1)$	$2\log_2 N$	$2\log_2 N$	$\sum_{i=0}^{u} 2\log_2 n_i$ + 2(u-1) ~2($\log_2 u$)	$1+\sum_{i=0}^{u} 1 = u+2$

表 6.2 金匙更新訊息數比較

		[9]	[10]	[11] with degree d	[12] , [8]	[13]	SSKM
Requested Subgroup Manager	Join	2	2			$\log_2 n_i$ +2	2 or 3
	Leave	n_i -1	n_i -1			$\log_2 n_i$	n_i or 0
Non-requested Subgroup Manager	Join	0	0			0	0
	Leave	0	1			1	1
Group Manager	Join		1	2($\log_d N$ -1)	$\log_2 N$ +2	(($\log_2 u$ +2)~ u) or 1	4 or 1
	Leave		u	d($\log_d N$ -1)	$\log_2 N$	1 or ($\log_2 u$ ~(u-1))	1 or u