國立交通大學

資訊科學系

碩 士 論 文

考慮路由器的廣播加密機制

A Broadcast Encryption Scheme Considering Routers

研 究 生：藍建宇

指導教授：曾文貴　教授

中 華 民 國　九 十 四　年 六 月

考慮路由器的廣播加密機制
A Broadcast Encryption Scheme Considering Routers

研 究 生：藍建宇　　Student：Jian-Yu Lan
指導教授：曾文貴 博士　　Advisor：Dr.Wen-Guey Tzeng

國 立 交 通 大 學
資 訊 科 學系
碩 士 論 文

A Thesis
Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

誌　　謝

在此感謝我的指導老師曾文貴教授，在我碩士班的學習過程中，不只在學業上帶領我走進密碼學的領域，更在生活和言行舉止上孜孜不倦的教導我，使我受益良多。另外，我要感謝口試委員，交大資工蔡錫均教授、交大資管劉敦仁教授與台科大資工吳宗成教授，在論文上給我許多良好的建議與指導，讓我的論文更加完善。除此之外，我要感謝實驗室同學，孝盈、政儒、與吳佑的幫忙，實驗室學長，成康的指導，以及實驗室學弟們在精神方面的鼓勵。

最後，我要感謝我的家人，不論在精神或物質上都給我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，謹以此文獻給所有我想要感謝的人。

考慮路由器的廣播加密機制

學生：藍建宇　　指導教授：曾文貴　博士

國立交通大學資訊科學系

摘要

廣播加密機制(broadcast encryption)是一種讓管理中心可以有效的廣播數位內容給廣大使用者的加密機制，使得只有非註銷使用者才有能力解開內容。廣播加密機制有廣泛的應用，例如在網際網路或行動無線網路上廣播電影、新聞、以及付費電視。

廣播加密機制分成有狀態型(stateful)與無狀態型(stateless)，無狀態型的機制雖然需要較長的廣播訊息複雜度(message complexity)，卻可以避免使用者必須參與全部金鑰更新過程這樣的條件。在無狀態型廣播加密機制中，目前最為可行的是子集差別方法(subset difference)。

在本篇演算法中，我們研究子集差別方法，發覺此方法所找出的覆蓋集合中，有些覆蓋是較無效率的。而實際的廣播加密應用上，並不是所有的非法使用者都一定要馬上被註銷其權限。我們提出新的方法，使得訊息複雜度比 SD 少了約 1/3，並且在適當路由器配合下，我們的方法也可以達到完美的註銷，使得所有非法使用者都無法解開訊息。

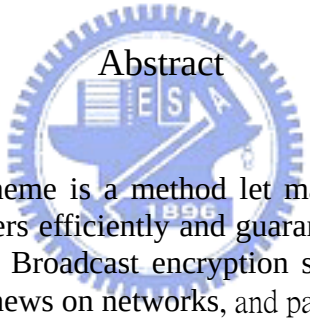
關鍵詞：廣播加密機制、無狀態型、有狀態性、訊息複雜度、子集差別方法、

A Broadcast Encryption Scheme Considering Routers

Student: Jian-Yu Lan Advisor: Dr. Wen-Guey Tzeng

Institute of Computer and Information Science

National Chiao Tung University



Abstract

Broadcast encryption scheme is a method let manager center broadcast digital content to large number of users efficiently and guarantees that only legal users have the ability to get the content. Broadcast encryption schemes have vast applications such as broadcasting movies, news on networks, and paid TV.

There are stateful and stateless broadcast encryption schemes. Although stateless schemes need more message complexity, they don't need users to keep online all the time. Among all of the stateless broadcast encryption schemes, subset difference method is the most practical.

In this paper, we researched subset difference method, and found some inefficient covers. In the practical applications, we don't need to revoke all illegal users immediately. Thus, we propose a new method that reduces its message complexity to 2/3 of SD's. Our method can have the perfect revocation in the help of special routers.

Key Words: broadcast encryption scheme, stateless, stateful, message complexity, subset difference method.

目次

第一章 引言

廣播加密機制是一種讓管理中心可以有效的廣播數位內容給廣大使用者的加密機制，使得只有非註銷使用者才有能力解開內容。廣播加密機制有廣泛的應用，例如在網際網路或行動無線網路上廣播電影、新聞、以及付費電視。

在廣播加密機制中，一開始的時候，管理中心分配給每一個始用者u一些金鑰K，K_u稱爲u的使用者金鑰集合。根據使用者的金鑰集合在之後會不會做更新，可以分做有狀態加密機制跟無狀態加密機制。訊息加密金鑰，SK，是用來加密所要廣播的訊息的金鑰，要廣播一個訊息M的時候，管理中心會用訊息加密金鑰(session key)SK對訊息M作加密，在廣播加密過的訊息密文時，會把SK的加密密文以及給使用者的加密SK時所用的金鑰資訊一起廣播，而型成下面格式的廣播密文:

＜有關資訊的標頭；$E_{SK}(M)$＞

$E_{SK}(M)$是用SK對M作加密後的密文。每一位非註銷使用者在收到廣播訊息後，必須利用定義好的演算法F，來計算F(K_u，有關訊息的標頭)=SK，再利用SK來解開$E_{SK}(M)$。但是對於任何一個被註銷的使用者來說，F(K_u，訊息的標頭)是不能夠得到SK的。另外，外了註銷使用者間防止互相合作的情形，即使擁有所有非註銷使用者的金鑰集合，也不應該有任何演算法可以在多項式時間內解開SK。

1.1 研究動機

廣播加密機制在日常生活中有著實際的應用，以付費的電視頻道來看，每個要收看電視節目的使用者必須和有線電視業者簽署合約繳付收看電視節目的費用，如果使用者沒有繳付費用，便會被撤銷收看電視節目的資格。此外，就算沒有繳付費用的使用者互相合作也沒有辦法收看電視節目。廣播加密機制可以滿足上述有線電視業者的需求，將電視廣播的訊息傳送給簽約收看電視節目的使用

者，而不被沒有簽署合約的用戶收看。

1.2 研究重點

在廣播加密機制中，直覺上最簡單的方法有兩個，第一個方法，是每個使用者跟管理者都共同分享一把金鑰。要廣播訊息時，管理者用每個合法使用者的金鑰來對資料加密金鑰 SK 作加密，把這些密文與要廣播的密文一起送出去，在這種情況，使用者只需要儲存一把金鑰，是最少的情形，但廣播訊息時，卻需要 N-r 個訊息長度，這是最多的情況。

第二個情況是，每個可能的使用者集合都有相對應的金鑰，每個使用者儲存他所存在的可能集合所對應的金鑰。要廣播訊息時，管理者根據當時的使用者集合，找出相對應的金鑰對 SK 作加密，在這種情況下，不管任何時候，廣播訊息只需要一個訊息長度，是最小的情況，但卻需要儲存 2^{n-1} 把金鑰，這在任何一種應用情況中，都是無法接受的。

因此，一個廣播加密機制所要考量的因素，除了必須滿足安全上的需求外，還得有廣播時，合理的加密 SK 所需要的訊息長度，我們稱此爲訊息複雜度；以及合理的使用者事先儲存的金鑰數量，我們稱此爲金鑰複雜度。

另一個重要性較前兩者來的輕，但也不可以太誇張的差的考慮要素是，使用者要要解開訊息加密金鑰 SK 所需的計算時間，我們稱此爲計算複雜度。一個廣播加密機制，除了要想辦法將上面三個要素做最佳化之外，一定要保持著安全上的特性，因爲通常廣播加密機制的安全性都是架構在底層所使用的加密演算法，所以廣播加密機制要負責的是，在加密演算法的安全支持下，確保不合權限的使用者無法得到管理中心加密 SK 所使用的金鑰。

第一個可行的廣播加密機制是 Noar 等人在 2001 年所發表的 SD 演算法[1]，在 2002 時 Halevi 跟 Shamir 對 SD 做了改良，提出了 LSD 演算法[3]。在 n 個使用者及 r 各被註銷使用者時，SD 的訊息複雜度是 2r，而儲存複雜度是 $O\left(\log^2 n\right)$，而 LSD 所需的訊息複雜度是 4r，儲存複雜度是 $O\left(\log^{\frac{3}{2}} n\right)$。在 2004 年，Goodrich 等人用另種方法對 SD 作改良[4]，使得儲存複雜度降爲 $O(\log n)$.

1.3 研究成果

在本篇演算法中，我們研究子集差別方法，發覺此方法所找出的覆蓋集合中，有些覆蓋是較無效率的。而實際的廣播加密應用上，並不是所有的非法使用者都一定要馬上被註銷其權限，這是在有狀態型機制中，早就有的概念[8]，尤其在使用者移動性高的無狀態型機制的應用，更是如此。因此，我們提出新的方法，使得訊息複雜度比 SD 少了約 1/3，並且在適當路由器配合下，我們的方法也可以達到完美的註銷，使得所有非法使用者都無法解開訊息。

1.4 各章節介紹

我們在第二章，會對在廣播加密的相關研究作個說明，介紹有名演算法的成果，在第三章，我們會詳細介紹 SD 演算法，因爲我們的演算法是根據 SD 作實作情況上的改良，在第四章，我們會介紹本篇的演算法，並且分析此演算法的效率。

第二章　相關研究

我們將在本章節對廣播加密機制的相關研究做介紹，首先會介紹有狀態與無狀態的差別，接下來會介紹許多廣播加密中，較爲突出的方法。

2.1 關於有狀態與無狀態

廣播加密機制從使用者儲存加密金鑰的方式來說，分爲兩種模式，一種是有狀態形式(stateful)，另一種則是無狀態形式(stateless)。兩種的差別在於，在某一時期，使用者與管理中心所溝通出來的金鑰，是否會影響下一時期所要溝通的金鑰。有狀態的廣播加密機制下，使用者能不能解開某一時期的金鑰更新訊息，會跟他是否擁有前一個時期的金鑰有關，最簡單的例子是，每一次變更，管理中心會給每個合法使用者足夠能力解開下一時期的資料加密金鑰所需的金鑰加密金鑰的集合，在這種情況下，非法使用者因爲沒有得到屬於他的金鑰更新訊息，就會失去解開下一時期訊息的資訊，而達到控管的功能。而無狀態的機制則是相反，通常是假設在使用者端，用以保存解密金鑰的儲存裝置是不能做變更的，例如快閃記憶體的裝置，使用者端不能保留前一時期溝通所得到的訊息，也沒有保存此訊息的必要，只要是合法的使用者，他們所擁有的儲存裝置已經擁有足夠資訊來解開金鑰更新的訊息，最常見的方法是，每個使用者各自擁有屬於自己的一個解密金鑰集合，在每一次的更新時，管理中心必須從所有的解密金鑰中挑選出一些解密金鑰來，這些挑選出來的金鑰，不能被任何一個已註銷的使用者所擁有，而這些金鑰足夠服務所有合法的使用者，在這種情況下，不管之前更新了哪些資料加密金鑰，在任何時期，永遠只有適當的金鑰加密金鑰會被挑選出來，任何沒有權限的使用者，將永遠無法解開金鑰更新訊息。底下章節，我們將會先介紹一個著名的有狀態廣播加密機制，接下來會介紹幾個有名的無狀態演算法。因爲本篇論文的演算法算是改良無狀態機制中註明的子集差別演算法而來，接下來的一章，我們會詳細介紹子集差別演算法。

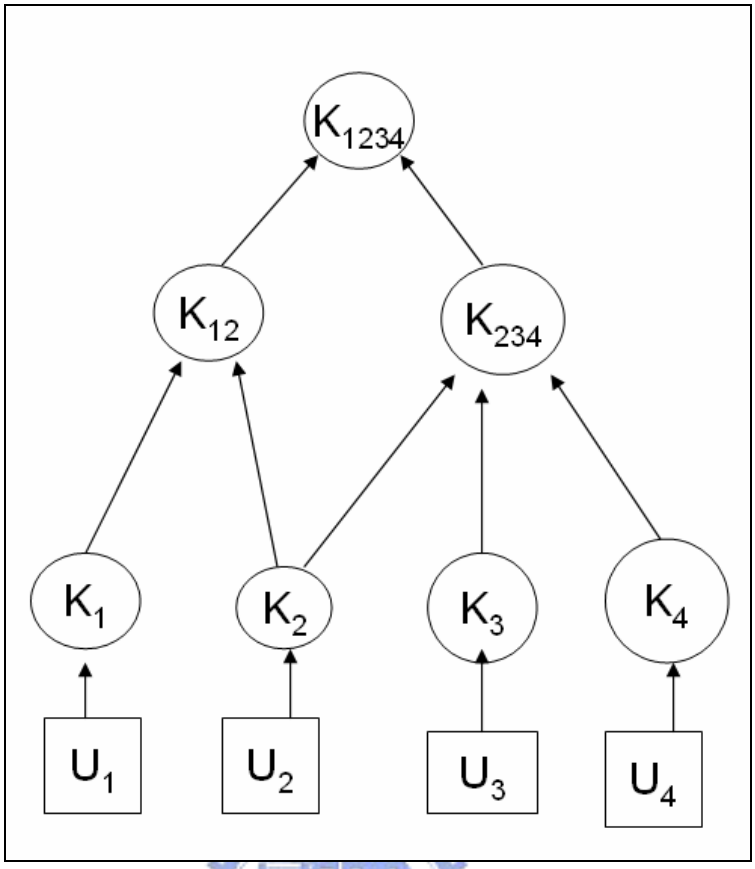
2.2 Wong 等人提出的邏輯金鑰樹方法

在有狀態型機制中，最爲著名的就是邏輯金鑰數架構，在 2000 年時，相當多的學者發表有關邏輯金鑰樹的相關方法，例如[2][5]，在這一小節對邏輯金鑰樹(logical key hierarchical)做一個一般性討論。

2.2.1 架構金鑰分配

不管在有狀態型機制或無狀態型，通常管理中心會建構一個圖形架構，將所管轄的使用者配對到架構中的一個元素，再討論金鑰的分配與維持。在邏輯金鑰樹的方法中，管理中心會保持一棵樹狀架構，在構成此樹狀架構的節點中，再分成兩種不同的節點，分別是代表金鑰的k-節點與代表使用者的u-節點，從管理中心所維持的一棵邏輯架構樹我們可以瞭解整個系統的三件事情，目前所參與的使用者集合U，目前所維持的有效金鑰集合K，以及U與K之間的關連性集合R，舉例來說，在下頁圖表 1 所表示的是在某一個時期，管理中心所維持的一個邏輯金鑰樹架構，在這樣一個架構中，$U=\{u_1,u_2,u_3,u_4\}$，$K=\{k_1,k_2,k_3,k_4,k_{12},k_{234},k_{1234}\}$是這個時期管理中心將要用來使用的金鑰資源，以及

$R=\{(u_1,k_1),(u_1,k_{12}),(u_1,k_{1234})(u_2,k_2),(u_2,k_{12}),(u_2,k_{234}),(u_2,k_{1234}),(u_3,k_3),(u_4,k_4),(u_3,k_3),(u_3,k_{234}),(u_3,k_{1234}),(u_4,k_4),(u_4,k_{234}),(u_4,k_{1234})\}$是目前狀態的關係，其中的$k_{abc...d}$的索引值似乎已經說明了哪些使用者擁有這把金鑰，卻無法實際發揮效果，真正提供管理中心維持架構所需資訊的，還得靠關係集合R的幫忙。



圖表 1.邏輯金鑰樹

2.2.2 使用者的新增與註銷

有狀態型機制跟無狀態型機制的其中一個不同點，是無狀態型機制不討論使用者的新增，在場景的假設中，無狀態型機制通常是管理中心在將金鑰集合分配給使用者時，就已經將一個超過目前使用者的適當數量 N 決定好，並且將該分配的金鑰提前用特定技術置入儲存裝置中。因爲在無狀態機制下，我們是假設使用者無法在後來的時段中，新增或變更所擁有的金鑰集合，若不事先將尚未用到的金鑰分配儲存好，新增的使用者是完全無法加入系統中。在無狀態型機制中，若有新的使用者加入，在實際的操作上，是由管理中心將已經事先處理好的金鑰儲存裝置直接提供給新使用者。在有狀態型機制就完全相反，既然有能力對更新的訊息加以儲存，系統就不需在完全不需要的情況下，考慮多餘的使用者，因爲當考慮進多餘的使用者時，就必須去維持一個更爲龐大的邏輯架構，會有一個更大的使用者數量 N，而在接下來我們會看到，有狀態型架構中的訊息複雜度，常會跟使用者數量有關係。

儘管邏輯金鑰架構根據圖學上的可能性，可以有星狀、樹狀、完全連結等不同架構，根據學者的研究分析，最爲有效的架構還是樹狀型結構。

我們先從下面圖表 2，對使用者的新增與註銷，將會對整個架構有何麼影響作一個一般性討論。在這個架構中長方形節點代表u-節點，圓形節點爲k-節點，有方向性的連結表示關係r，透過方向性連結，u-節點u_i所能夠旅行的所有k-節點，就是在此時期下，u_i所擁有的金鑰集合。圖示中的兩個架構，分別是一個擁有八個使用者的時期架構跟一個擁有九位使用者時期的架構，圖中顏色較深的k-節點是多出來的那位使用者所擁有的k-節點，也是他會影響到的所有k-節點，當故事的進行是從上面走向下面，也就是新增u_9時，管理中心必須爲整個架構做小變更，除了新增一個u-節點外，還得把所有之前擁有跟u_9一樣的k-節點的所有使用者，做該有的更新，在例子中，管理中心得把新的金鑰$k_{1\text{-}9}$跟k_{789}傳給u_7跟u_8，把$k_{1\text{-}9}$傳給u_1到u_6，若不新建一個新的k-node，直接用原本的$k_{1\text{-}8}$與k_{78}，這表示u_9可能有能力解開以前的廣播訊息，這是一個危險的狀況，在更新新的金鑰時，因爲管理中心跟使用者間，已經有一些建構好的金鑰存在，可以用這些舊的金鑰對新的金鑰做加密，例如，用k_{78}對{$k_{1\text{-}9}$，k_{789}}做加密，再傳給u_7與u_8。

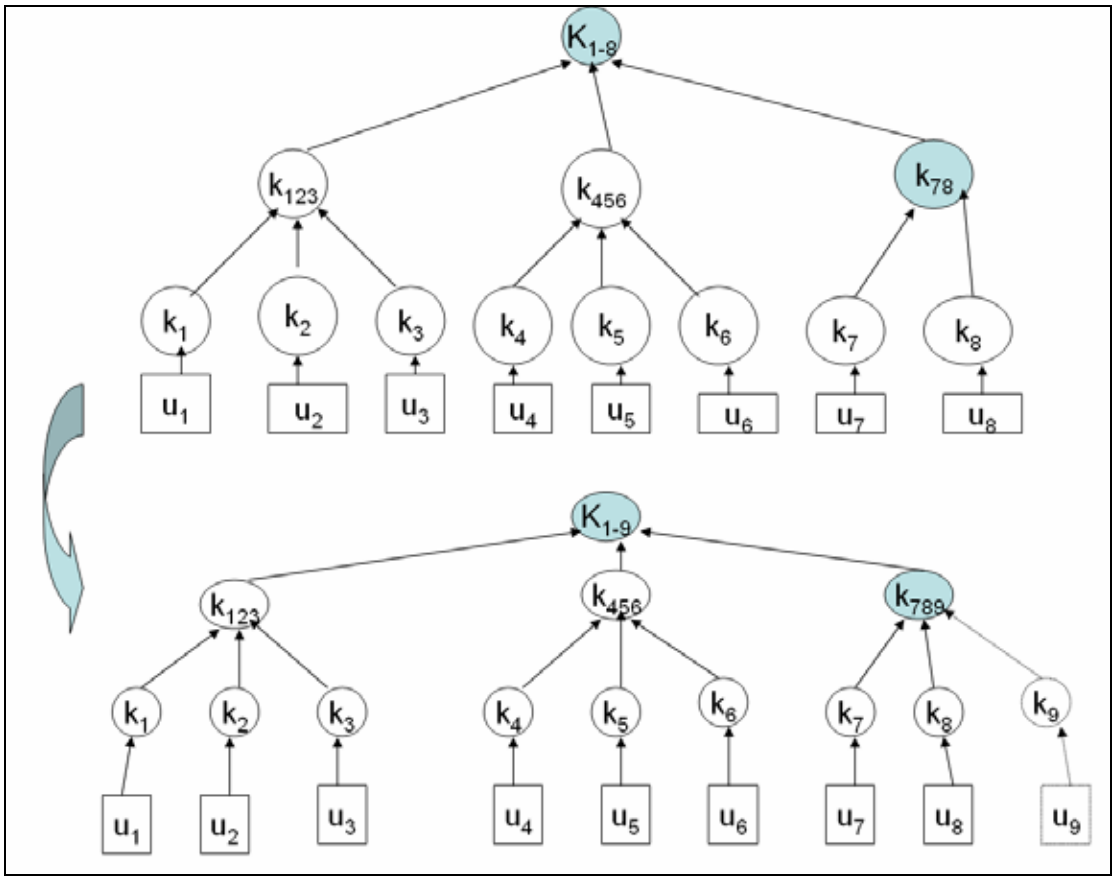
當故事由下往上進行，也就是註銷u9 時，因爲$k_{1\text{-}9}$與k_{789}都是u_9所擁有的資訊，必須跟換新的金鑰來確保安全，因此整個架構中，我們必須重新傳送$k_{1\text{-}8}$跟k_{78}給所有該更新的使用者。

雖然所要做的事情就是更新該換掉的金鑰給該拿到的使用者，但實際用來完成這項任務的方法，卻可分成使用者導向、金鑰導向、跟群體導向，以新增u_9爲例子來看，管理中心可以針對u_7，用k_7加密{$k_{1\text{-}9}$,k_{789}}，再用k_8加密相同的{$k_{1\text{-}9}$,k_{789}}，等等從使用者爲導向做更新的動作，也可以針對$k_{1\text{-}8}$，加密$k_{1\text{-}9}$來服務給原本擁有$k_{1\text{-}8}$的使用者，再用k_{78}加密k_{789}，這種根據哪些k-node有變化，就加密哪些密文的金鑰導向型更新動作，管理中心也可以把用$k_{1\text{-}8}$加密$k_{1\text{-}9}$，k_{78}加密k_{789}再把所有訊息廣播給{u_1，u_2…u_8}，這樣的作法，對某些使用者來說，有很多部分的訊息是沒有意義的，但在可以充分發揮廣播效能的情況下，這樣的處理方使可以省下很多訊息。

有狀態型機制中，儘管有不同架構不同導向的處理方式，所有訊息的瓶頸都發生再註銷使用者時，這是因爲很多有效率的金鑰變成不能使用的緣故，例如，在新增的情況，$k_{1\text{-}8}$是很多使用者都擁有的，善用這把金鑰可以省下很多訊

息，但在註銷的情形是，因爲u_9也擁有這把金鑰，所以就不能用它來加密，於是得用共用性比較少的金鑰，例如k_{123}、k_{456}來代替。

研究指出，對於管理中心來說，群體導向的更新方式是比較有效的，而對於使用者來說，使用者導向的更新方式是最有效的，而金鑰導向的更新方式則適合用在有些特定小頻寬的網路中。不管哪一種更新方式，訊息複雜度約爲 $O(\log N)$。



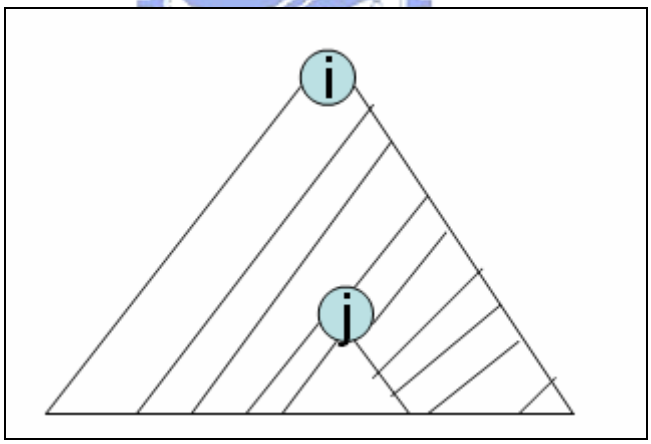
圖表 2.上面的圖表是八個使用者時所需要的邏輯金鑰樹,下面的圖表示九個使用者所需要的邏輯金鑰樹架構

2.3 子集差別的相關方法

在無狀態的廣播加密機制中，影響這門研究最大的就是 Naor 等人在 crypto 2001 上發表的子集差別方法，因爲下一章節我們將詳細說明子集差別演算法，在此我們簡單說明子集差別演算法採用的子集合，以便說明其他跟子集差別演算法有關的方法。

2.3.1 Naor 等人提出的子集差別方法(SD)

在子集差別方法中，採用的是完全二元樹的架構，在這個架構中，使用者是被放置在最底下的葉節點，這個演算法所採用的子集合是以$S_{i,j}$來定義，代表意思是，在此集合中的節點，都是i節點的後代，而且，都不是j節點的後代。這樣的實際意義可以從下圖圖表 3 得到感覺。

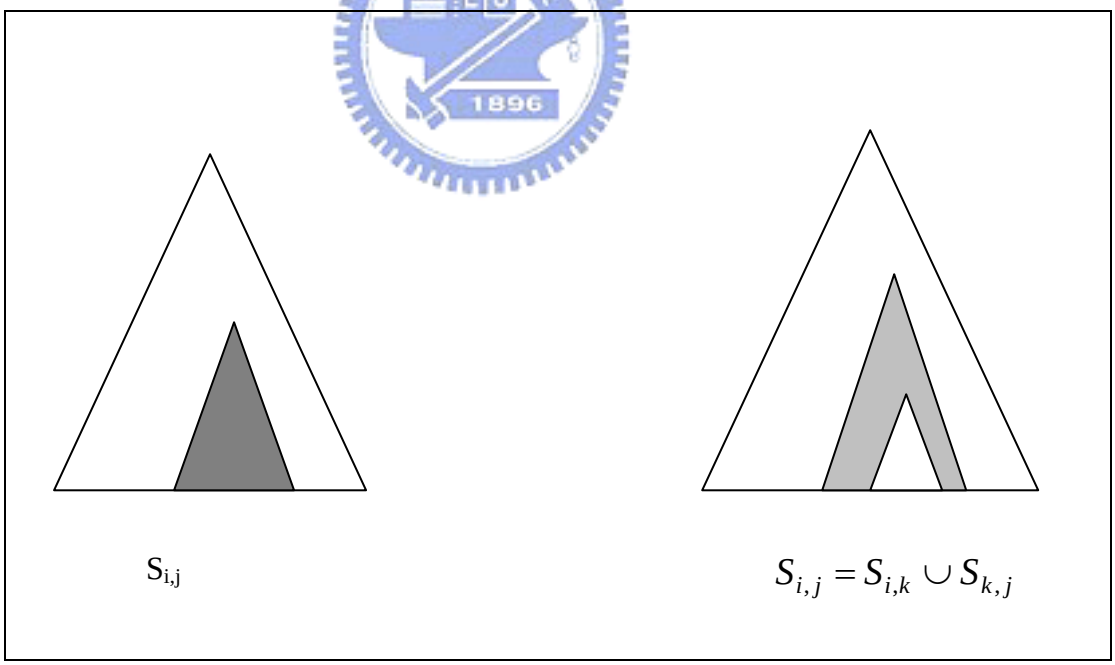


圖表 3.$S_{i,j}$在圖形上的意義

在一個N個使用者的完全二元樹中，總共有$\log_2 N$層的中間節點，藉著在使用者端儲存屬於這些子集合的金鑰$K_{i,j}$，子集差別方法提供一個儲存量、訊息量、以及計算量都相當不錯的金鑰更新方式。使用者需要儲存$O\left(\log^2(n)\right)$的金鑰，而訊息量的最大值爲 2r-1。

2.3.2 Shamir 等人提出的層級子集差別方法(LSD)

在子集差別方法中，曾經用圖表 4 左邊的圖示來說明$S_{i,j}$，其中的白色區域就是$S_{i,j}$，用相似的方法，我們以右邊的圖來說明層級子集差別方法中所定義的子集合。在這個演算法中，將子集差別方法的子集合切爲兩塊，如圖所說明的，利用$S_{i,j} = S_{i,k} \cup S_{k,j}$這樣性質，完全分離開來。舉例來說，假設有個使用者m在SD方法中是位於$S_{i,j}$中的，但在層級子集差別方法中，$S_{i,j}$被切成兩個子集合，他被歸在$S_{k,j}$中了，因爲層級子集差別方法的刻意安排，同樣的，原本在子集差別方法中，m也有可能位於$S_{l,j}$中，但在層級子集差別方法裡面，他又被歸於$S_{k,j}$中，如此一來，在子集差別方法方法中，本來m得儲存這兩把金鑰(在下一章，我們會說明，使用者儲存的其實是$S_{i,j}$的標籤，而不是金鑰)，但在層級子集差別方法中，只要記憶一個$k_{k,j}$就好，利用這樣的方法，層級子集差別方法把所需儲存的金鑰減少爲$O\left(\log^{\frac{3}{2}}(n)\right)$，但是所需的最大訊息量因爲子集合的縮小而增加爲 4r-2。



圖表 4.子集差別方法與層級子集差別方法所定義的子集合

2.4 Tomoyuki 所提出的冪集合演算法

Tomoyuki 在 2002 年的 asiacrypt 發表了新的演算法[6]，用計算量換取儲存空間與傳輸訊息量，這就是冪集合演算法(power set method)，這個方法需要比子集差別方法來的有效多的儲存效率，所犧牲的代價是使用者在計算出 SK 時，需要花費較多的計算時間。

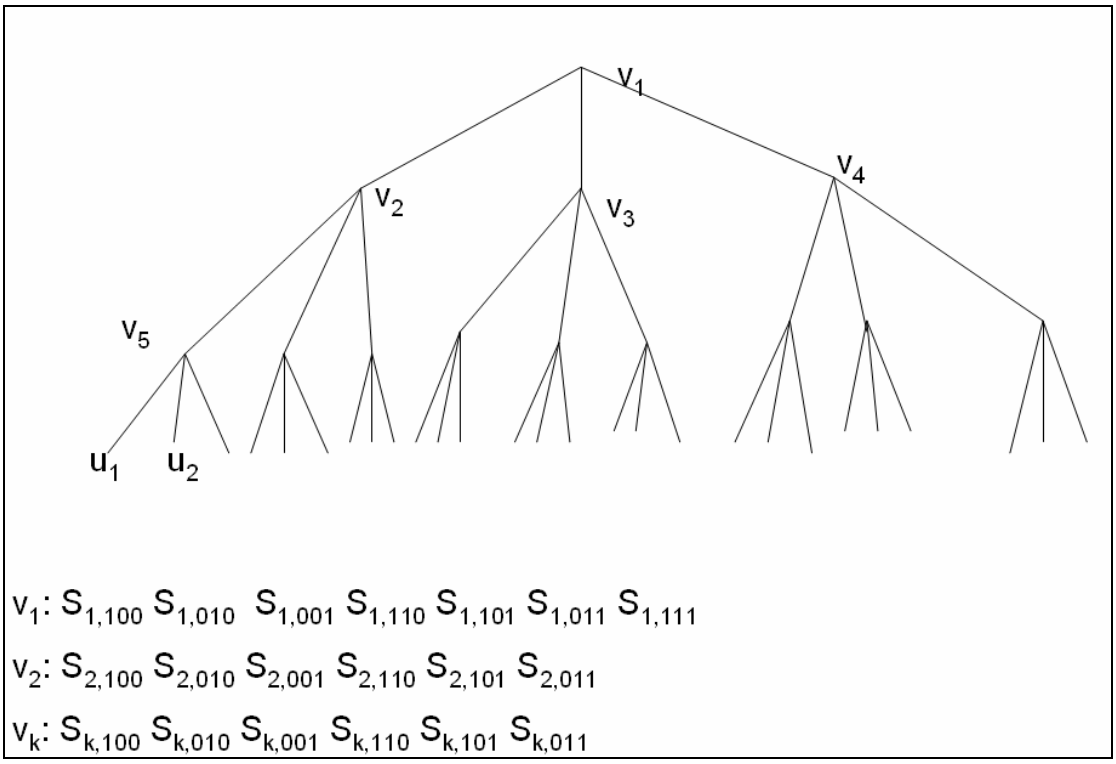
2.4.1 架構金鑰分配

跟一般的無狀態行機制一樣，冪集合演算法首先定義一群子集合，屬於此子集合的使用者就會分配到一把金鑰，假設冪集合裡面所建構的樹中，每個中間節點擁有a個子節點(degree)，在此演算法所定義的子集合是以 $S_{k,b_1b_2...b_a}$ 來表示，其中第一個k是表示這個集合是以從根節點開始寬先次序第k個中間節點來做考慮的點集合，而接下來的$b_1b_2...b_a$則是表示a個bit，用來表示 $S_{k,b_1b_2...b_a}$ 這個集合實際所包含的子樹，當b_i=1 時，這表示包含V_k節點從左邊數來第i個子節點，也就是說，每個子集合都是針對中間點來考慮，考慮他下面a 棵子樹得所有包含情形，並且排除掉全部爲 0 跟全部爲 1 的情形，共有 2^a-2 個情形，對於根節點V1，我們多加一個子集合$S_{1,11..1}$，以下面圖表 5 的例子來說明，這是一個a=3 的結構樹，全部有 $\frac{N-1}{a-1}$ 個中間節點V_k，當V_k=V_2， $S_{2,100}$ 所包含的元素就是以{V_5}這樣的集合， $S_{2,101}$則是{V_5,V_7}的集合。當管理中心決定完所有架構後，要爲每一個$S_{k,B}$， $B=b_1b_2...b_a$ ，安排一個質數$P_{k,B}$，並且公布出去，接下來要從 Z_M^* ，而M是兩個大質數q_1，q_2相乘所得，M=q_1q_2，隨機挑出整個系統的常數K，有了這些 $\left(2^a-2\right)\frac{N-1}{a-1}+1$ 個屬於各個集合的質數$P_{k,B}$跟K，就決定了屬於每個集合的金鑰$SK_{k,B}$，先把所有的質數$P_{k,B}$相乘得到T，則每個集合的金鑰

$SK_{k,B}=K^{\frac{T}{P_{k,B}}} \bmod M$ ，根據每個中間節點所處的集合，管理中心分配這些集

合的金鑰給此節點，例如，分配給V_5 $SK_{2,100}$、$SK_{2,101}$、跟$SK_{2,110}$這三把金鑰，因爲我們多定義了$S_{1,11\cdots1}$這個集合，所以根節點的每個子節點都額外分配到$SK_{1,11\cdots1}$。

定義完這些參數後，管理中心要分配每一個使用者u_j一把金鑰MK_j，首先我們考慮從根節點V_1到葉節點u_j的路徑，總共會經過$\log_a N$個中間節點，這些中間節點總共被分配到$\left(2^{a-1}-1\right)\log_a N+1$把子集質數，把這些子集的$P_{k,B}$全部相乘起來得到屬於$u_j$的$w_j$，則使用者$u_j$所分配到的金鑰$MK_j = K^{\frac{T}{w_j}} \bmod M$，例如，$MK_1$就是用$P_{1,100}$、$P_{1,110}$、$P_{1,101}$、$P_{1,111}$、$P_{2,100}$、$P_{2,110}$、$P_{2,101}$、$P_{5,100}$、$P_{5,110}$、$P_{5,101}$這些子集質數所得到的$w_1$加以運算而來。



圖表 5.中間節點的金鑰

2.4.2 加密與解密

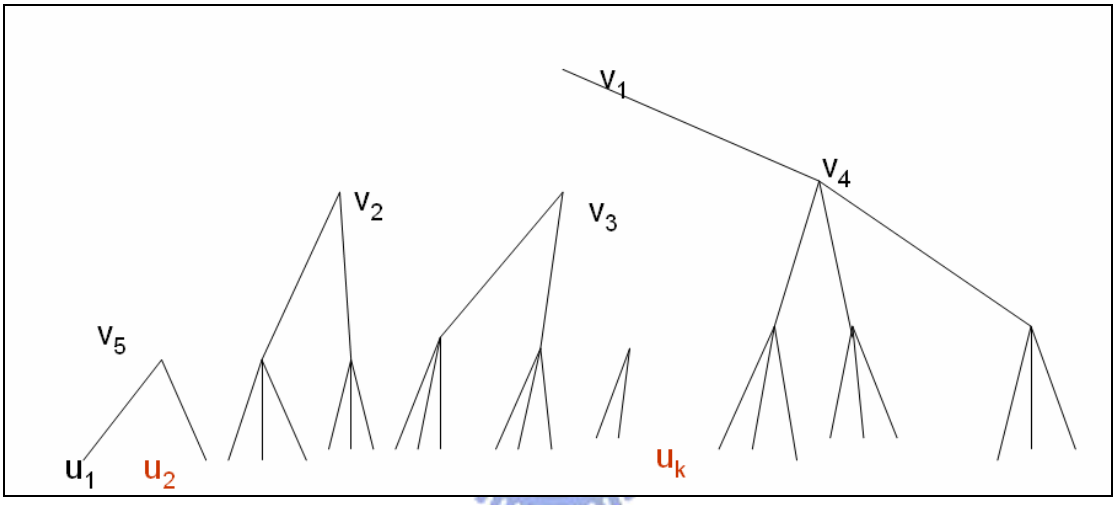
當管理中心要發送一個金鑰更新訊息時，他必須先針對目前架構數中被註銷使用者分佈的情況，決定要使用哪些加密金鑰來加密資料加密金鑰，管理中心決定的方法非常簡單，就是把被註銷的使用者u_j從結構樹上拔除，這樣一個動作，會使得從u_j到根節點所有路徑上的連結消失，舉例來說，假如u_2跟另一個中間的使用者u_k被註銷權限，我們將被註銷者到根節點的路徑拔除，形成下面圖表

6 的結構，於是剩下了$S_{5,10^1}$、$S_{2,011}$、$S_{3,110}$、…這幾個子集合，管理中心就使用屬於這些子集合的金鑰$SK_{5,10^1}$、$SK_{2,011}$、$SK_{3,110}$、…來加密金鑰更新訊息，

當使用者收到訊息後，必須先找出屬於自己的金鑰加密密文，接著必須計算出用來加密這段密文的金鑰$SK_{k,B}$，接下來利用

$$MK_j^{\frac{w_j}{Pk,B}} \bmod M = \left(K^{\frac{T}{w_j}} \right)^{\frac{w_j}{P_{k,B}}} \bmod M = K^{\frac{T}{P_{k,B}}} \bmod M = SK_{k,B}$$

使用者必須從公開的眾多$P_{k,B}$中計算出w_j，就可以接著算出$SK_{k,B}$。



圖表 6.拔除註銷使用者到根節點所經過的路徑

2.4.3 效率

冪集合演算法最大的優點是使用者只要儲存一把金鑰，他的訊息複雜度一樣是跟r成比例，約爲 $r\left(\frac{\log\left(\frac{N}{r}\right)}{a}+1\right)$ 的訊息量，當r佔N的一定份量時，所需的訊息跟子集差別演算法相近，而最大的缺點是需要較多的計算量，使用者每次計算$SK_{k,B}$時，需要 $\left(2^{a-1}-1\right)\log_a N$ 個模數相乘跟一個模數次方運算，這樣的計算量，跟其他演算法比起來，是非常可觀的，當使用者端的處理平台是一個對耗電

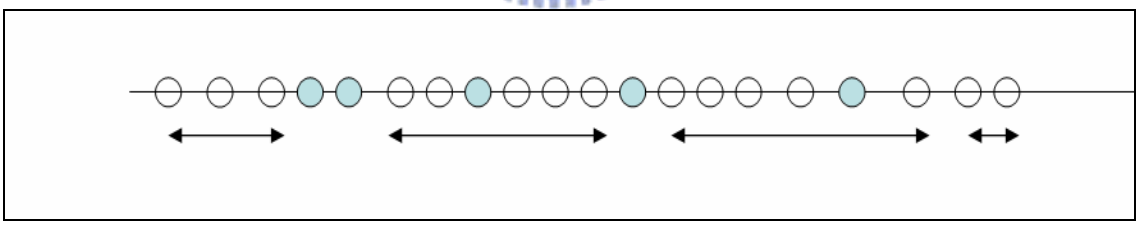
量相當重視的環境時，是非常不適合的。

2.5 區間穿刺方法

在 2005 年的 Eurocrypt，Nam-su Jho 等人發表了一篇新的廣播加密方法[9]，這個方法的子集合定義與子集差別方法大有不同，是近年來新的突破，這個方法定義了許多可以彈性調整的參數，經過作者的實驗，在相當多的應用情況上，都較子集差別方法來的有效率。

2.5.1 子集合定義 $(p, c)-\pi$

在這個方法中所定義的子集合較爲彈性。所用的子集合是以(p,c)區間來表示，說的是一個 p 穿刺 c 區間的區段，這個區段中，最多有 c 個使用者，此區段的開始與結尾都是合法使用者，而且中間最多只能有 p 個非法使用者。我們以下面圖表 7 說明：



圖表 7.(1,6)區間的例子

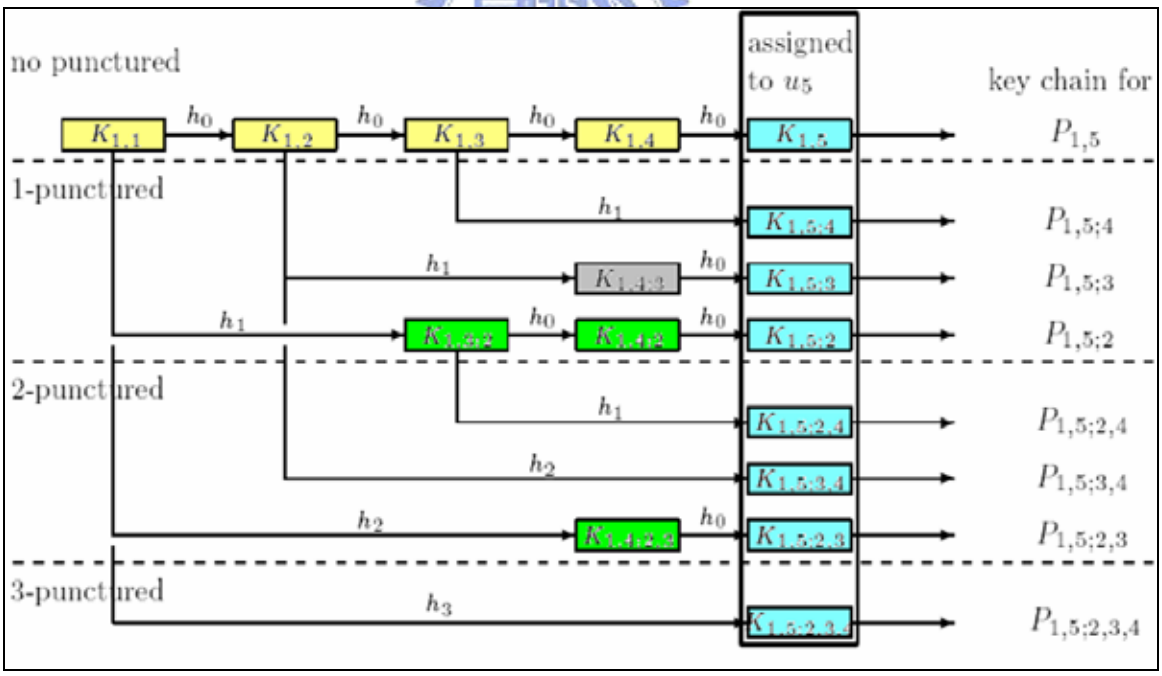
上圖是一些(1,6)區段的例子，第二個跟第三個區段都是可以達到最大量，六個，的區段，而當中的非法使用者也都不超過一個，屬於定義範圍，至於第一個區段，因爲右邊有連續兩個非法使用者，要是想含括到第六個合法使用者的話，就得覆蓋到這兩個非法者，與(1,6)的定義相違背，應此變成上圖所示的結果：第一個區段只有三個使用者。

定義了上述的子集合之後，管理中心就要分配每位使用者所需要儲存的金鑰。一個使用者 i，會有可能位居哪些(c,p)區間呢？首先我們從每一個區間的開始點 j 開始看，因爲一個區間不會有超過 c 個使用者，所以 $i-c+1<j<i$ ，

所以總共有 c-1 個可能。

在這個方法中，利用了很多不可逆函式(one-way function)，使得擁有某些金鑰的使用者，可以利用不可逆函式，繼續往下推算，卻無法推算出此金鑰的反函數。在這一類的應用中，通常會幫每一個點k分配一個最初始的值K_k，靠這個初始值，擁有此金鑰的使用者可以利用公開的不可逆函式 $f(\bullet)$，推算出$F(K_k)$，甚至$f(f(K_k))$等其他含數值，在這種應用下，通常使用者k至少會拿到K_k，假如要讓k的下一位使用者k+1 也可以做出$f(K_k)$，卻不想把完全只屬於k的祕密金鑰K_k給其他人知道的話，我們可以直接給使用者k+1 一個已經算好的值$f(K_k)$，如此一來，使用者k+1 幾乎擁有跟使用者k一樣的計算能力，也可以計算$f(f(K_k))$等其他值。

在區間穿刺方法裡面，就是利用上面所說的技巧安排給使用者的金鑰，只是稍微複雜了些，不只定義一個不可逆函式，而是需要h_0、h_1、…h_p等p個不可逆雜湊函式，其中每個雜湊函式的數字下標i表示h_i這個函式是當上一個金鑰到下一個金鑰中間經過多少個註銷使用者，我們利用圖表 8 來說明此概念，並藉此說明一位使用者必須擁有哪些金鑰。



圖表 8.當c=5 時，u5 所從$K_{i,1}$推導而來的金鑰

這個圖所表示的是使用者u5 所必須儲存的金鑰，首先管理中心分配給每一位使用者i一把金鑰$K_{i,1}$，從每一把金鑰可以用眾多的雜湊函式得到$K_{i,j}$，而當一把金鑰要從$K_{i,j}$變成$K_{i,j+1}$或$K_{i,j+2}$時，這分別表示j+1 到j中間沒有註銷使用者($K_{i,j+1}$的例子)以及

j+2 到j中間有一個註銷使用者($K_{i,j+2}$的例子)。因此，我們會利用h_0函式對$K_{i,j}$作用來得到$K_{i,j+1}$，以及利用h_1來得到$K_{i,j+2}$，例如，在上面圖中，我們利用$K_{1,3}$經過h_1得到$K_{1,5;4}$，(“;”後面的數字表示從 1 到 5,經過了哪些註銷使用者)，利用這樣的概念，因爲 u5 有可能位於由u1 開始的區段中，而我們又不想把最直接的$K_{1,1}$給u5，因此，我們必須把從u1 開始到u5 中間有可能的每一個區端的金鑰，事先算好，並且分配給u5，如上圖所示，接著當考慮從u2 開始時，又有很多區段的金鑰要事先分配給u5，如此一直重複到u4 開頭的區端金鑰爲值。

2.5.2 尋找覆蓋的方法

穿刺區間方法式將所有使用者都安排在一條直線上，並且分配每一個使用者從左往右遞增的標示順序，當我們要在其中一個時期，對現有的非法使用者做註銷時，就是根據目前狀態，利用貪婪演算法，對整個使用者的直線做切割，切出非常多的(c,p)區間來，而屬於這些區間的金鑰，就用來加密資料加密金鑰 SK。

利用之前所介紹的表示方式，假設管理中心使用了$P_{i,j;x_1,...,x_q}$這個區段的金鑰來加密，這時一位合法的使用者 uk，$i < k < j$，收到訊息以後，他要利用現有的金鑰來解開用$P_{i,j;x_1,...,x_q}$加密的密文，他只要先找到自己的索引值 k，到底位於 i 到 j 的哪個位置，從 i 到 k 經過哪些註銷使用者，他就可以找到適合的金鑰，並且利用已經公開的不可逆雜湊函式，計算出$P_{i,j;x_1,...,x_q}$的金鑰來。

2.5.3 效律

這個方法會需要的訊息複雜度，就是在任一個時期，有可能切出來的最多數量的(p,c)區間數，作者利用歸納法證明了當有 N 個使用者及 r 個使用者時，一

個(p,c)-π方法最多可以切出$\left\lfloor \frac{r}{P+1} \right\rfloor + \left\lceil \frac{N-(p+2)\lfloor r/(p+1) \rfloor}{c} \right\rceil$個區間來，而所需要的儲存複雜度是$\sum_{0}^{p+!} \binom{c-1}{k}$，這樣的方法，只要被註銷使用者的數量大於$\frac{N}{2c}$時，就會比子集差別方法來的有效率，作者還另外提出對非法者較少的情況做出改良的方法。

第三章 子集差別法演算法

因爲我們的方法是改進子集差別方法而來，爲了下一章解釋本篇論文演算法時的需要，我們在這一章，對子集差別方法做更詳細的說明。

3.1 子集合的定義

在子集差別方法中，採用的是完全二元樹的架構，在這個架構中，使用者是被放置在最底下的葉節點，這個演算法所採用的子集合是以$S_{i,j}$來定義，代表意思是，在此集合中的節點，都是i節點的後代，而且，都不是j節點的後代。整個結構樹的高度是$\log_2 N$。管理中心爲結構樹中不是葉節點的中間節點i分配一個m位元長的標籤，這些標籤是隨機且獨立選取的。管理中心再選擇一個隨機產生器(pseudo random generator)G，在給G m個位元長度的輸入下，G會輸出 3m 位元長度的資料，我們用這樣的G，來爲以i爲根節點的子樹中的節點j，定義他的標籤。假設管理中心分配給i節點的標籤爲x， x的長度是m，利用G，我們可以得到 3m 長度的G(x)，把G(x)切成三等份，從左到右的三份m長度的值，分別拿來當作:分配給$S_{i,j}$的金鑰$K_{i,j}$，給i的左子節點k的標籤，給i的右子節點l的標籤，因此,每一個中間節點，根據他的祖先節點得到不同的標籤，也可以根據這些標籤得到屬於此節點的不同金鑰。而從一個i節點的標籤，可以定義出i底下任一個後代節點j的標籤,進而定義出$K_{i,j}$，所以一開始管理中心只要隨機選出分配給每個節點的標籤，就可以定義出所有的金鑰。而這種用一個不可逆函式來幫助產生金要的方式，有一種特性:假設j'是j的一個後代節點，在擁有$K_{i,j}$情況下，可以得到$K_{i,j'}$，但要因此得到其他金鑰(不同的i，或者j'不是j的後代節點)，卻是很難的。利用這個特性，可以省下記憶複雜度，因爲有一部分的金鑰可以靠G產生出來。

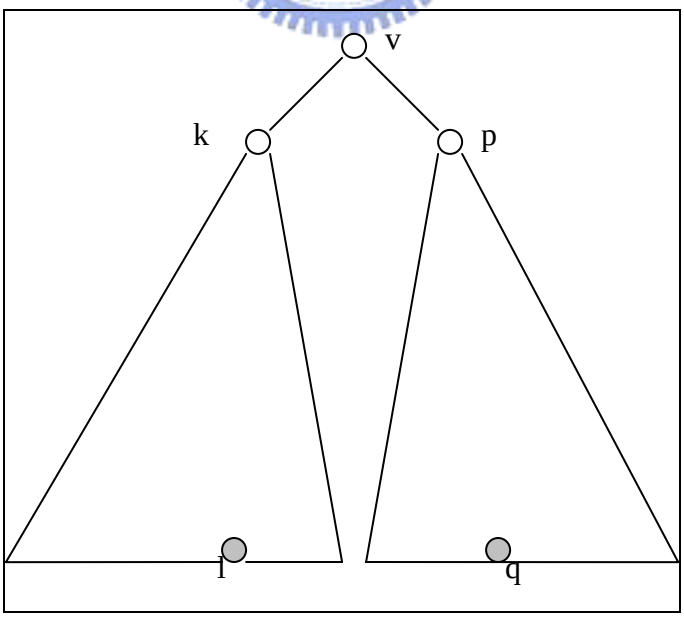
每個使用者必須能夠得到他所屬的每個子集合$S_{i,j}$的金鑰，因爲此演算法利用G來推導相關金鑰的關係，使用者並不需要儲存所有的金鑰，下面一節我們會對所需的儲存複雜度以及溝通複雜度作分析。

3.2 覆蓋的定義及尋找覆蓋

管理中心在處理好金鑰分配後，如同一般無狀態型機制一樣，把處理好前置工作的金鑰儲存裝置分配給使用者，就完成了新增使用者的工作，這裡我們討論如何註銷使用者。

當管理中心要送出訊息時，必須考慮目前 r 個註銷使用者的分佈，找出適當的 m 個子集合，使得沒有任一個非法的使用者屬於這些集合，而因爲合法使用者分別屬於所選出的其中一個子集合中，我們稱這些被選出來的子集合爲覆蓋(cover)，因爲這些選出來的子集合必須能夠包含所有合法使用者的緣故。我們稱尋找這些子集的動作爲尋找覆蓋。

一開始管理中心擁有一個覆蓋集合，用以收集找到的子集合，當結構樹上還有兩個以上的註銷使用者時，管理中心必須找出同時包含任意兩個註銷使用者的最小子樹，我們假設這棵子樹的根節點爲v，v的左子節點爲k，右子節點爲p，而所包含的兩個註銷使用者中，左邊爲l，右邊爲q，將$S_{k,l}$與$S_{p,q}$增加到覆蓋集合中(假如k=l或是p=q，則代表所對應的集合是空集合，就不需要加進覆蓋中)，



圖表 9.圖示表示註銷方法所找到的最小子樹，以及各節點名稱

接著把v標示爲註銷節點，並且刪除v以下的所有節點，重複這樣的步驟，直到我們將整個結構樹的根節點r標示爲註銷節點，或是整棵樹只剩下一個註銷節點v爲

止，在第二個情況的話，我們要將$S_{r,v}$加進覆蓋集合中。

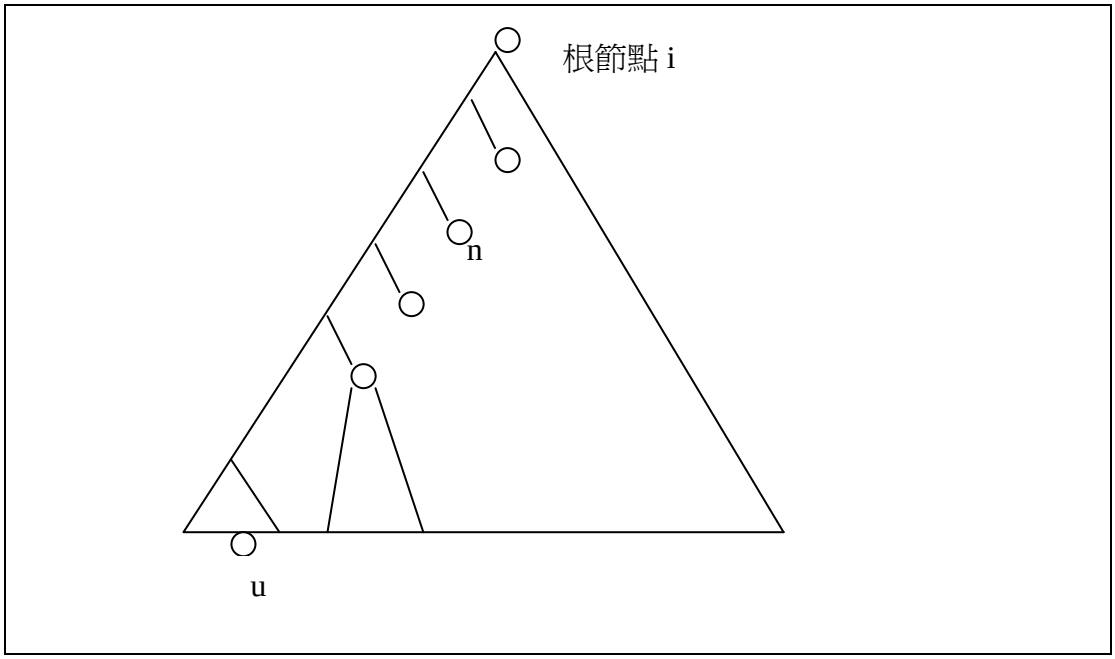
整個尋找覆蓋的流程結束後，覆蓋集合中的所有子集合，就代表我們要拿來做加密的相對應金鑰，因此，需要多少個子集合來覆蓋所有使用者，也就表示會需要多大的訊息複雜度。

3.3 所需要的訊息與儲存複雜度

要討論一個使用者u需要儲存多少金鑰，可以從會有多少個有意義的子集合$S_{i,j}$包含這一個節點來看。假如沒有隨機產生器G的幫助，每一個使用者就要儲存$O(n)$的金鑰，但因爲在子集差別方法中利用了隨機產生器的關係，金鑰與金鑰間有一定的關係存在，所以使用者可以不用儲存那麼多的金鑰，以下圖來解釋，假設我們考慮所有包含使用者u以i爲根節點的的集合$S_{i,j}$，假設i的右子節點爲l，只要u擁有$S_{i,l}$的標籤，他就可以利用隨機產生器推導出右邊所有子集合的標籤與金鑰$K_{i,m}$，m是l的後代節點；相同的，只要u擁有$S_{i,n}$的標籤，任何i與n後代節點o的子集合$S_{i,o}$的標籤與金鑰，他都可以得到。因此，一個使用者u所需儲存的資訊，是所有u的祖先節點i，從i到u的路徑上的節點的另一方向子節點j所形成的$S_{i,j}$的標籤，如圖表 10 所示。如此說來，使用者u需儲存多少這樣的標籤呢？因爲整個結構樹上，u總共有$\log_2 n$個祖先節點，所以i的可能個數就是$\log_2 n$，從每個i到u的路徑上，最多會有$\log_2 n$，最少有 1 個垂掛的節點j，因此，u所需儲存的標籤數量就是

$$1+2+3+...+\log_2 n=\frac{\log_2 n(\log_2 n+1)}{2}$$

我們簡化所需的儲存量爲$O(\log^2 n)$。



圖表 10.圖示表示使用者u所需要儲存的$S_{i,j}$標籤，當i是根節點時的情形

接下來我們要討論廣播時所需的訊息複雜度，在廣播的訊息中，包含管理中心選擇的覆蓋集合的訊息說明、用各個覆蓋集合的金鑰所加密的金鑰密文、以及用加密金鑰所加密的訊息密文，其中，覆蓋集合的訊息說明相較起來相當的短，所以在討論訊息複雜度時，我們變成討論管理中心用了多少的覆蓋集合。回顧之前討論的尋找覆蓋的方法，在每一個階段中，我們最多新增兩個覆蓋集合，並且刪除一個註銷節點，所以在最後一步之前，我們最多新增了 2r-2 個覆蓋集合，而在最後一步，我們頂多新增一個覆蓋集合，因此最壞情況，會需要 2r-1 個訊息。

第四章 考慮特殊路由器下對子集差別方法的改良

經過上一章節對相關研究的介紹，以及對子集差別方法的認識，我們將開始介紹本篇論文所提出的演算法，首先我們會先從廣播訊息如何再網路上傳輸做一個說明，再介紹子集差別方法中較無效率的覆蓋情形，接著就是本篇的論文。

4.1 廣播的網路處理方式

在這一節，我們介紹在現行網路架構上，multicast 所使用的傳播方式。

4.1.1 host 端加入廣播群組

在 IP(Internet Protocol)位址中，有一部份的位址是保留給廣播群組用的，32 位元的 IP 位址，當高的四個位元爲 1110 時的位址，就是一個廣播用的位置，其值爲 224.0.0.0 到 255.255.255.255。當一個 host 想加入一個廣播群組時，必須利用 IGMP 協定(Internet Group Management Protocol)向路由器報告，申請加入某個群組，此後路由器若收到廣播給此群組的封包，就會廣播到子網域中。

4.1.2 路由器的傳播

當一個訊息由一個 host 傳送出去，當到了路由器時，路由器就需要決定出要將此封包傳送出去的路徑，有別於一般網路封包的傳送，在 multicast 時，路由器有可能將一個封包送達多個路由器。一個支援 multicast 的路由器除了跑 IGMP 協定讓主機可以註冊廣播群組外，還必須執行 Multicast Routing Protocol

例如 DVMRP 等，用以決定出所需傳算的路徑。

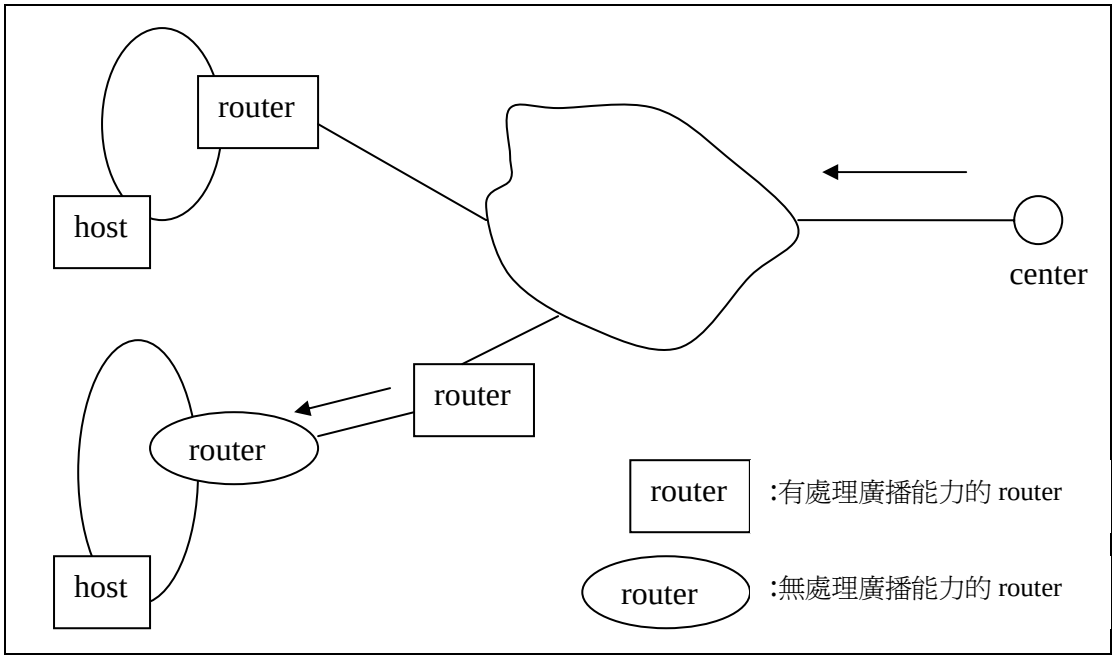
4.1.3 網路傳輸的特性

網路傳輸在硬體層與連結層上有一些特性，例如：每一個位元的值有一定的機率會因爲誤差而產生錯誤；參與的路由器越多，找尋路由路徑所要考慮的節點數量越龐大，演算法出現失誤或找出較差路由方式所發生的機率就越大；應用層所製造的訊息，在實際下放到連結層時，必須切成符合實際硬體規定的網路封包，資料量越大，切出來的封包越多；…等。在這些因素的引響下，在廣播者到接收者中間的漫長路途中，所能夠使用的頻寬會是實際用作時的瓶頸。相對的，我們可以把最後接段(last mile)的頻寬，視爲較不重要的因素，因此，縮小中間傳輸所需要的訊息量，將是一個不可忽視的改善。

假如有兩個廣播加密方法，他們所需的訊息複雜度，從使用者端看來也需一樣多，若是某一個方法可以減少必須在廣播者到接收端中間漫長路途的訊息量，那麼這個方法的訊息將較不容易出現意外，而以較快的時間到達接收端。

4.1.4 考慮特殊路由器下的網路特性

在廣播加密時，通常是由一個管理中心，將所要加密的有價值資料，用資料加密金鑰加密得到 $E_{SK}(M)$，再挑選適當的金鑰加密金鑰對資料加密金鑰 SK 加密後，廣播出去，訊息離開管理中心所在網域的路由器後，就開始在 internet 上旅行，各個支援 multicast 的路由器在接收到訊息後，判斷所管理的區域內是否有人加入此廣播群組，有的話就把廣播訊息傳送到區域網路中，而有些路由器並沒有支援 multicast 功能，無法管理底下有多少成員加入廣播群組等事情，必須由有支援此功能的路由器幫忙，代爲管理，並在收到廣播訊息時，將訊息的目的地設爲群組會員的網路位置，直接以一般網路封包傳送。

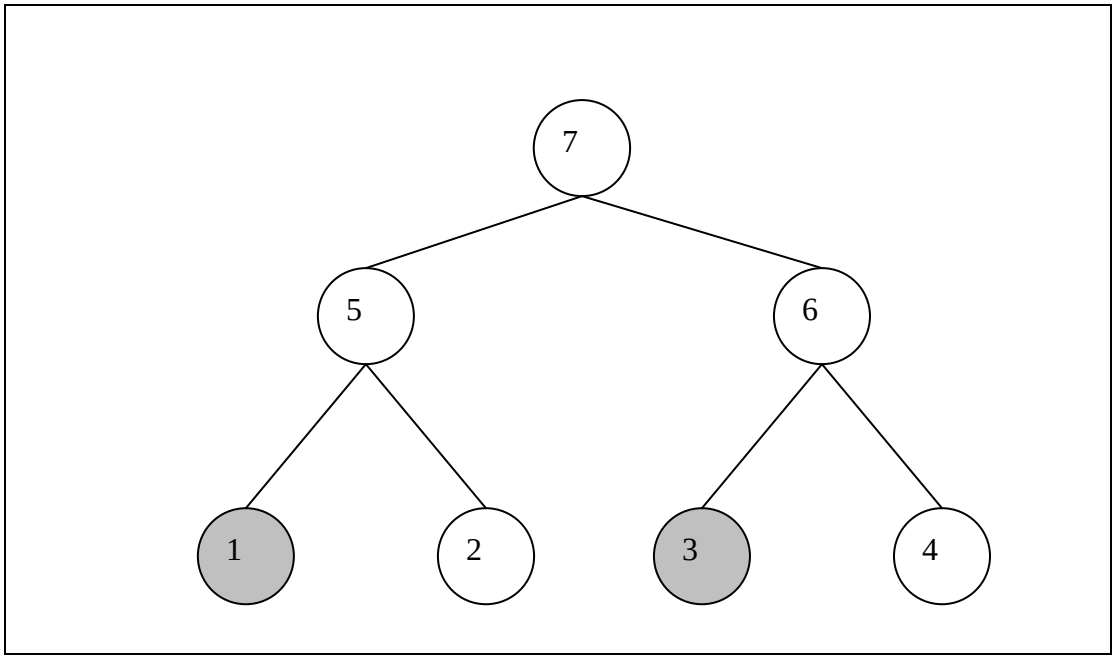


圖表 11.圖示解釋有處理廣播能力與無此能力的路由器，互相合作達到廣播功能的情形

當廣播訊息所要傳送的距離相當遠時，路途中所能得到的頻寬就就因共用者多而越小，也更有可能面臨前一小節所說的網路問題，我們就要對個網路區段的訊息量，做不同的考量，因爲同網域內的訊息，相較於網際網路上的訊息量來的少，頻寬又相對的來的大些，因此，一份一樣大的訊息在網際網路上旅行所要花費的時間跟可能造成的封包遺失的風險，遠較同網域內的傳送來的大，廣播加密所需的訊息量龐大，若能夠節省在網際網路上傳送的量，在子網域端即使所需的量變的稍大些，也是相當有價值的。

在本篇演算法中，我們假設所布設的路由器，可以將 IP 層的網路封包，重組成應用層的訊息，並且從廣播訊息中，挑選網域內用戶所需要的金鑰更新訊息，並對特定訊息，可以進行額外加密的能力。在這樣的情景下，我們提出一個改良子集差別演算法的方式，使得所需要在網際網路上傳送的訊息，能夠得到約 1/3 訊息量的改善。

4.2 較無效率的訊息



圖表 12.圖示爲子集差別方法中，會導致無效率覆蓋的情形

圖表 12 表示的是一個四個葉子的子樹,在此例子中，node1 跟node4 是被註銷的使用者，SD演算法爲了幫這個子數尋找適合的覆蓋，會找到$S_{5,1}$跟$S_{6,3}$這兩個子集合，所選出來的分別是$K_{5,1}$跟$K_{6,3}$這兩把金鑰，在這樣的例子中，一個被選出來的金鑰，其實只有提供一位合法使用者使用，這是演算法中，使用的最沒效率的更新訊息，在其他模式中，一個挑選出來用來加密的金鑰，都可以拿來給多個合法使用者使用，因此我們認爲這種註銷使用者分配模式是一種需要較多餘的訊息量的模式，我們定義這樣的模式爲保留性模式。

本篇論文所提出的演算法，就是根據保留行模式所改進的，把可處理加密訊息的路由器考慮進去的話，node1 與node2 有相當大的機會不會在同一個區域內,這表示若我們挑選一把較上層的金鑰加密金鑰，例如$K_{X,7}$，來加密所要分配的資料加密金鑰，再廣播到網路上，等到路由器接收到廣播的訊息，挑選所要向所屬網域廣播的加密訊息時，再作判斷，若此時node1 或node3 剛好與其中的合法使用者，node2 或node4，在同樣一個網域時，路由器就需要針對node1 及node3 作特別的加密處理，作出這樣的加密$E_{K2}(E_{Kx,7}(SEK))$訊息，如此的話，只有node2 可以解開這個訊息，取得資料加密金鑰，相反的，假如在這個區域內，只有node2 一位時，路由器就無需再作任何處理，可以直接把接收到的訊息向下廣播。

4.3 主要演算法

本演算法分成管理中心處理部分跟路由器處理部分，以下就兩方面所要處理的演算法，個別說明。

4.3.1 管理中心的處理

當目前集合中還有兩個以上的被註銷節點時，嘗試選擇一個同時包含任意兩個被註銷節點的最小子樹，我們假設這兩個被註銷節點爲r1 跟r2,這棵子樹的根節點是i，i的左子節點爲l，i的右子節點爲m，因爲r1 與r2 一定分別爲l及m的後代節點，在不失一般性下，我們假設r1 屬於以l爲根的子樹中，而r2 屬於以m爲根的子樹中，與子差別分法類似的，我們找出了$S_{l,r1}$與$S_{m,r2}$兩個集合，在子集差別的方法中，找到這兩個可覆蓋被註銷節點的集合後，便可以把相對應的金鑰加入到實際所要使用的金鑰集合中，但在這裡我們稍停一步多做處理。我們考慮l與r1 以及m與r2 分別差距多少層，假如其他一組的差距爲一層，假設爲l與r1，我們就把這組的祖先節點，也就是l，標示爲保留節點，在考慮完這兩個集合後，要根據下面三種情形作處理：

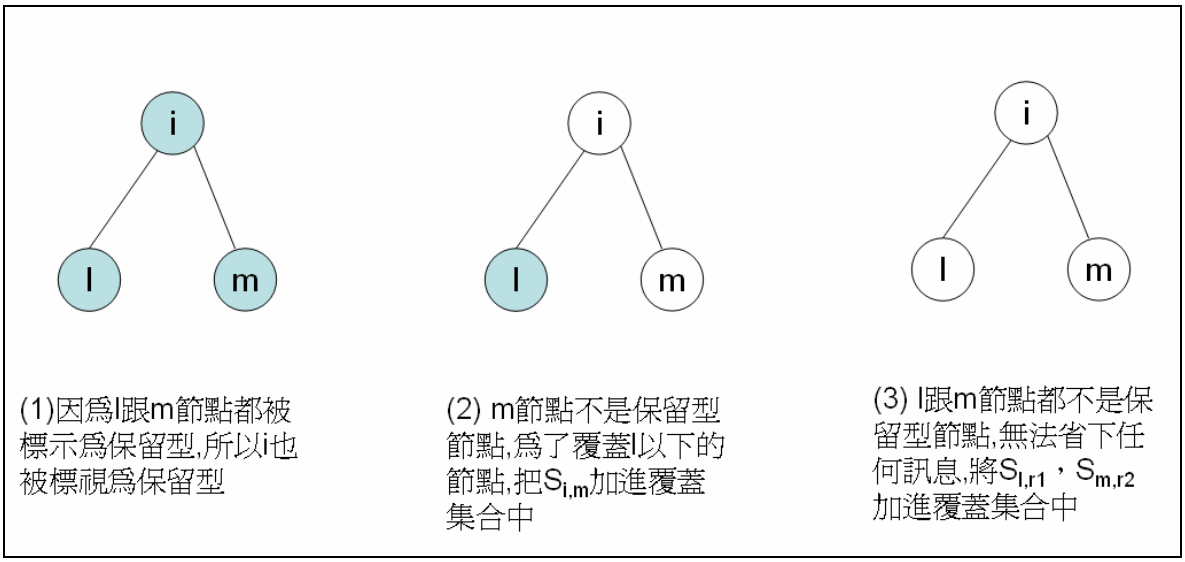
(1) 假如l跟m都被標示爲保留節點，這表示剛好滿足上一節所說的保留性模式，我們像子集差別方法一樣，拔除$S_{l,r1}$，$S_{m,r2}$兩集合的節點，但我們不將這兩個集合加入到覆蓋集合中，爲了可以讓保留的屬性往樹狀結構的更上層移動，我們將i節點標示爲保留節點，然後跟子集差別方法一樣，將i節點設爲被註銷節點。

(2) 假如l節點跟m節點兩個只有一各是保留節點的話，我們假設l爲保留節點，這表示$S_{m,r2}$所包含的節點不少，用這一個集合所屬的金鑰來加密的話，可以服務到很多合法節點，因此不能省略掉，所以我們把$S_{m,r2}$加入到覆蓋集合中，而另一邊因爲有保留節點，表示在之前的步驟中，有可能已經省略掉了一些集合，爲了使從l節點以下的所有合法節點都能夠有一個加密訊息服務，我們

把$S_{i,m}$加入到覆蓋集合中，相同的，拔除$S_{l,r1}$，$S_{m,r2}$兩集合的節點，並且將i節點設爲被註銷節點。

(3) 假如l節點與m節點都沒被標示爲保留，這種情形跟子集差別方法的情況一模一樣，只需將$S_{l,r1}$，$S_{m,r2}$加入覆蓋集合中，拔除$S_{l,r1}$，$S_{m,r2}$兩集合的節點，並且將i節點設爲被註銷節點就可以。

圖表 13 分別表示上面所說的三個情形:



圖表 13.圖示表示 l 與 m 的三種情形

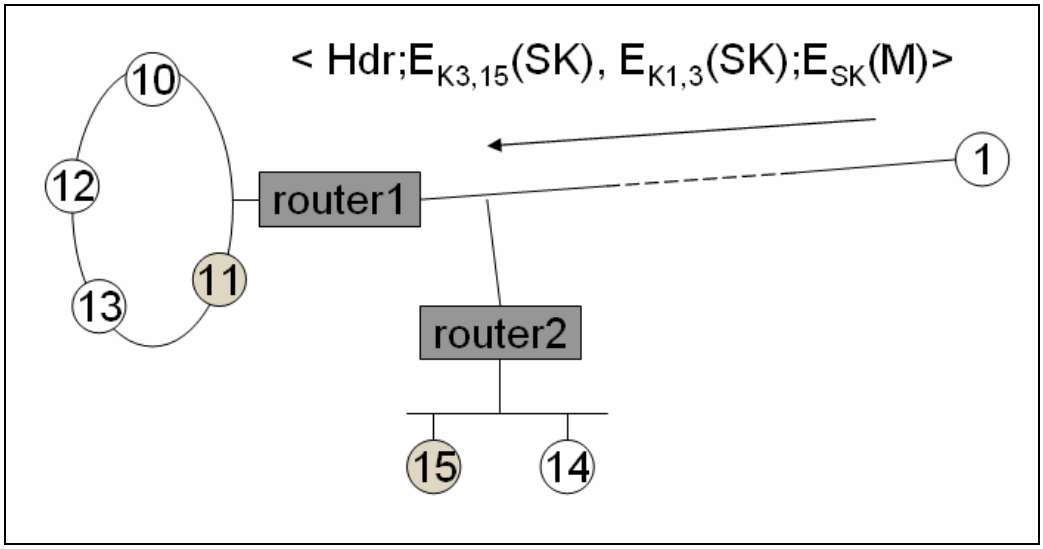
整個管理中心要做的就是不斷進行這樣的步驟，直到根節點變成被註銷節點，或是整棵樹只剩下一個被註銷節點爲止，接著將覆蓋集合中的集合所對應的金鑰加密金鑰，拿來對資料加密金鑰做加密，就可以廣播出去了。

4.3.2 路由器端的動作

本篇論文定義的路由器必須有能力知道所需服務的使用者中，有哪些使用者是合法使用者及哪些是已經被註銷的。在接收到管理中心送出的廣播訊息時，路由器要找從管理中心所使用的$S=\{S_1, S_2, S_3, \ldots, S_i\}$的覆蓋集合中，找出能使所有底下的合法使用者都能夠得到訊息的子集合$S'=\{S_1', S_2', \ldots, S_a'\}$，這當中會有些訊息某些被註銷使用者也有能力解開，所以路由器必須根據底下的註銷使用者，找出$S''=\{S_1'', S_2'', \ldots, S_b''\}$覆蓋所有底下的註銷使用者，而$S'''=S' \cap S''$就是有可能被註銷使用者解開的加密訊息，我們得將用這些金鑰做加密的密文用路由器與使用者共有的短時間金鑰再做一次加密。

以下面圖表 14 爲例，假設節點 1 是管理中心，將內容廣播出去。當訊息來到路由器 2 時，因爲他底下的群組成員中，只有節點 15 是被註銷的，而此使用者並沒有能力解開訊息，所以可以直接廣播到區域網路中；不同的是，路由器 1

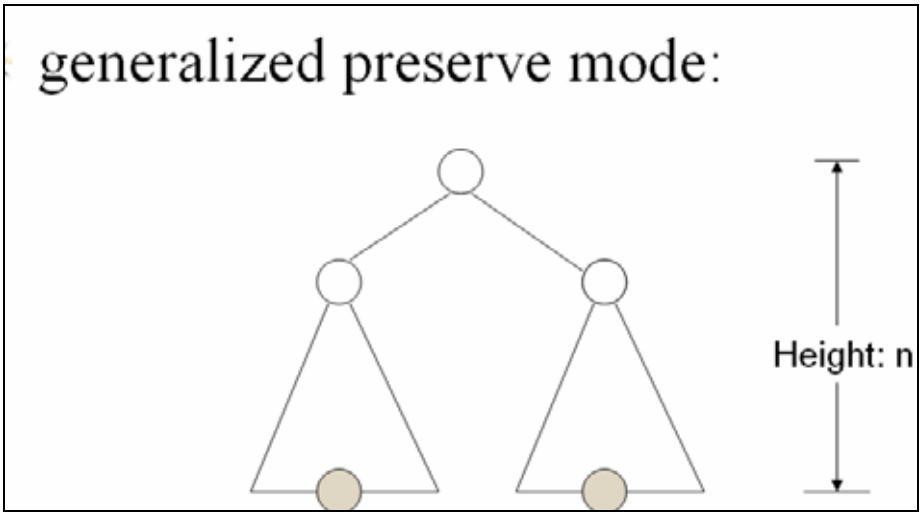
底下的群組成員中，因爲節點 11 是被註銷使用者，而他又有能力可以解開 $E_{K1,3}(SK)$，因此，我們需要路由器 1 幫我們再做另一次加密，製作出$E_{k10}(E_{K1,3}(SK))$再廣播到區域網路中。



圖表 14.圖示中,節點 1 爲管理者，把所選擇的覆蓋加密金鑰後廣播出去，當兩個不同網域的路由器收到訊息後，所需採取的動作將不一樣

4.4 更一般性的取捨

在我們的演算法中，都是根據是否符合保留性模式，來決定是否標示爲保留性節點，進一步影響是否要省下加密訊息。接下來我們提出更一般性的模式，所謂的更一般性模式是相對於之前的保留性模式而言，前面一節所提的保留性模式都是針對一個兩層高的子樹做討論，討論這棵子樹的四個子節點，被註銷節點的分佈情況，是否滿足構成此子樹的兩棵單層子樹，都各有一個被註銷節點，而這節要說的更一般性模式，就是把保留性模式的子樹，從兩層高一般化爲N層高。一個n層高子樹的 2^n個葉節點中，n≦N，只要被註銷節點的分佈滿足構成此子樹的兩棵n-1 層子樹，都各有一個被註銷節點的話，這棵n層高的樹就是保留型模式，就有可能因此省下訊息。



圖表 15.一般化的保留模式

在我們的一般性演算法中，因爲只要有任一個滿足保留性模式的子樹，其高度小於 N，這棵子樹的根節點就會被標示爲保留性節點，因此，同樣的一個註銷型節點分佈情況下，一個內部節點被標示爲註銷型節點的機會，在一般性的演算法下將比基本型演算法來的高，而一般型演算法實際可以節省下的廣播訊息量，也較基本型來的大，隨著 N 的變大，可以節省的訊息將會變多；從一個保留型子樹所覆蓋的被註銷的使用者看來，從管理中心加密出來的訊息中，有一個訊息是他可以解開的，在一般性模式中，跟這個被註銷使用者在同一覆蓋的合法使用者相對的較多些，代表這個訊息會被廣播在更多網域內，當我們考慮的路由器有可能幫非法使用者舞弊，沒有誠實的將應該另外加密的訊息作加密，就直接廣播出去的話，這名非法使用者就有更高的機會可以解開資料加密金鑰，當然，如果我們假設路由器都處於正常運作模式，這項缺點是可以忽略的。

在這一節，考慮 N 的取捨下，我們提出新的一般性模式，可以節省更多的訊息，但在路由器有可能做出非法行爲時，必須承擔非法使用者有更高機會，竊得金鑰的風險。

4.5 效率分析

在這一節，我們將考慮所提出的演算法，跟子集差別法相比，可以節省多少訊息量。因爲基本模式架構較簡單，比較好分析，我們先從基本模式討論起。

假設目前使用者數量爲 N，被註銷使用者數量爲 R，在這樣的場景下，爲了簡化分析方式，我們說一個使用者節點有 R/N 的機率會是被註銷的使用者節點。

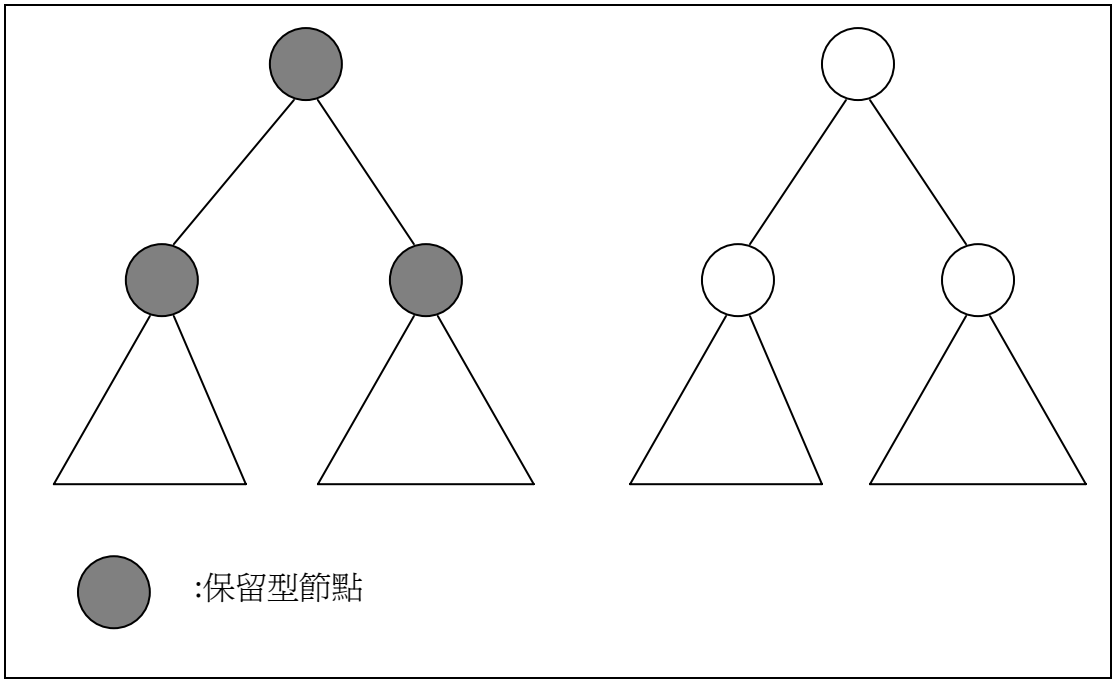
4.5.1 名稱定義

Xn 是一個有 n 個葉節點的子樹，Rn 是一個有 n 個葉節點，並且滿足保留型模式的子樹的所有可能情形所構成的集合，Yn 表示一個有 n 個葉節點的子樹，在基本模式可以省下來的訊息量。

4.5.2 可節省訊息量的期望值

這一小節，我們提出一個估計 Yn 期望值的通式，配合其他的邊界狀況，可以算出在 N 個使用者，當中有 R 個被註銷使用者情況下，基本模式可以從子集差別方法節省下的訊息量。

首先我們考慮一個有 n 個葉節點的子樹，分成兩個情況來討論，一個是根節點的兩個子節點都是保留型節點的情況，另一個情況則是相反。在第二個情況下，Xn 可以節省的訊息量就是兩棵以根節點的兩個子節點的爲根節點的子樹，所能夠節省的訊息量的和。在第一個情況時，根節點也會變成一個保留型節點，在這種情況下，只會用一個集合來覆蓋所有的點,而惟一會造成這種情況，只有在最底層每一個兩層高的子樹全都符合保留型模式，才有可能，所以全部有 N/2 個被註銷使用者，在子集差別演算法下，本來會需要 N/2 個訊息，所以可以省下 (N/2 - 1)個訊息量。



圖表 16.上圖表示考慮根節點及其兩個子節點的兩種可能情形，從左到右分別是上述的狀況一以及狀況二。

根據上面分析不同情況下可以節省的訊息量，我們可以得到下列通式：

$$E(Y_n) = (\frac{n}{2} - 1) \times \Pr[Xn \in Rn] + 2 \times E(Y_{\frac{n}{2}}) - 2 \times (\frac{n}{4} - 1) \times \Pr[Xn \in Rn]$$

有了上面通式，我們還得考慮兩個週邊情況，對於E(Yn)來說，能夠省下訊息的最基本必須是四個葉節點以上的子樹，可以節省一個訊息，所以：$E(Y_4)=\Pr[X_4 \in R_4]$，這個式子的意思是，有四個葉節點的子樹，所能夠節省下的訊息的期望值，就是這棵子樹會是保留型模式的機率，另外，從之前的狀況一我們也可以了解，一個有n個葉節點的子樹會Xn會滿足保留型模式只有在其下兩個子節點全都是保留型節點，因此： $\Pr[Xn \in Rn] = \Pr[X_{n/2} \in R_{n/2}]^2$ ，而

$\Pr[X_4 \in Rn] = 4(\frac{R}{N})^2 \times (1 - \frac{R}{N})^2$ ，我們令 $\Pr[X_4 \in R_4] = P$ ，於是

$E(Y_4) = P$

$E(Y_8) = P^2+2P$

$E(Y_{16}) = P^4+2P^2+4P$

….

$$E(Y_N) = P^{\frac{N}{4}} + 2P^{\frac{N}{8}} + ... + \frac{N}{8}P^2 + \frac{N}{4}P$$

我們可以用微積分對上式夾擊得到簡化式子，但因爲積分後的式子也是很複雜，一樣不容易分析，但根據模擬結果，因爲 P＜0，當 N 夠大時，上式的前面項都是非常小的，所以我們只需考慮最後面三項就可以得到跟模擬趨近的期望值，因此，我們得到 $E(Y_N) = \frac{N}{4}P + \frac{N}{8}P^2 + \frac{N}{16}P^4$

4.6 與之前子集差別方法之比較及貢獻

因爲我們的演算法是在能夠處理特殊廣播訊息的路由器的情況下，對子集差別演算法所做的改良形式，因此我們將所提出的演算法跟子集差別演算法作效率上的比較，比較的方式是模擬當有 N 個使用者及 R 個被註銷使用者時，我們的演算法跟子集差別演算法所需要的訊息量。

4.6.1 模擬說明

在我們的模擬中，我們在選定的 N 情形下，模擬當裡面有 R 個被註銷使用者的分佈情形共 T 次，在 T 次實驗中，R 個被註銷者是平均且隨機的分佈在使用者中，初始化分佈情形後，再實際模擬子集差別演算法跟我們所提出的演算法，得出不同演算法所需要的訊息量，在進行完 T 次模擬後，將每個演算法所需要的訊息量平均起來，就是最後的結果。依據模擬的結果，因爲我們的實驗次數 T 足夠大的關係，我們可以看到平均所需的訊息量，與上節所分析的期望值非常相近，可以佐證我們在上一節所做的效率分析。

4.6.2 模擬結果

下面圖表 17 是當有 N = 10000，T = 400 時，我們針對不同的 R，所模擬出

來的結果，第一行的 revoked_num 是被註銷者數量 R 的值，第二行 SD_message 是子集差別演算法在特定 R 時，所模擬出來平均的訊息量，接下來的 my_message，基本演算法所需的平均訊息量，以及當 n=3 時演算法所需要的平均訊息量，跟 n=4 演算法所需的平均訊息量。

revoked_num	SD_message	my_message	my3_message	my4_message
1000	1110	1028	1023	995
1501	1564	1395	1387	1322
2002	1948	1679	1663	1566
2503	2270	1892	1861	1744
3004	2524	2043	1992	1871
3505	2717	2142	2067	1953
4006	2841	2196	2102	2010
4507	2905	2213	2111	2034

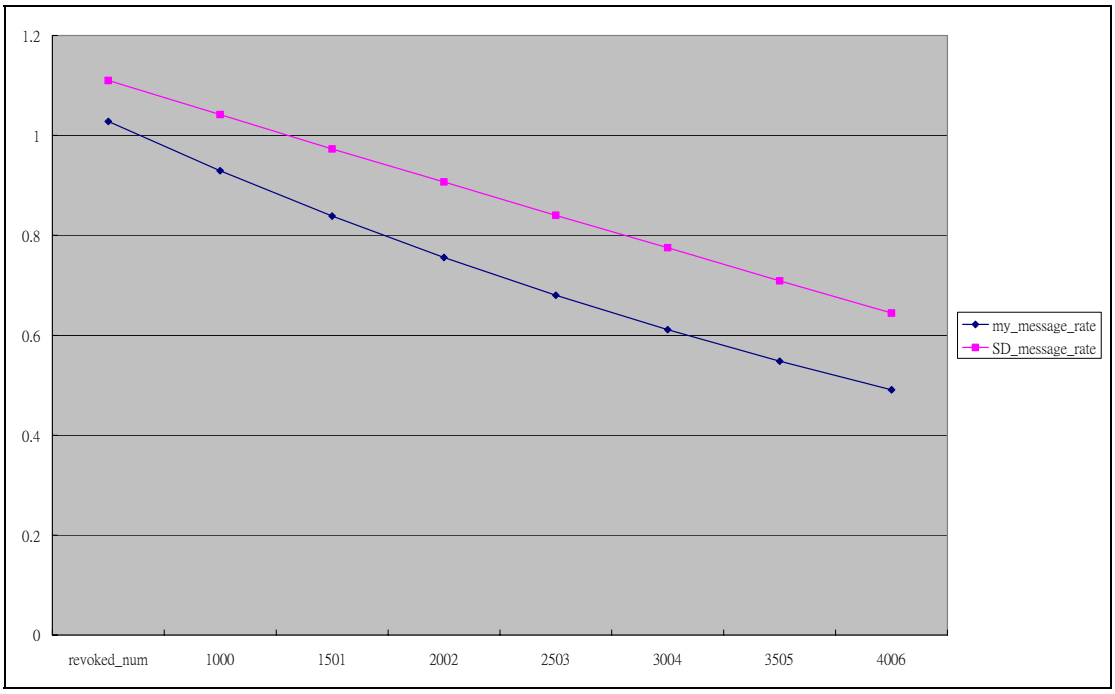
圖表 17.N=1000，T=400 時，我們的演算法與子集差別方法所需要的訊息量

接下來的圖表 18，說明的是不同 R 時，所提出的演算法所節省的訊息，所佔的百分比，從實驗中我們可以發現，當被註銷者數量越接近 N/2 時，所能夠節省的訊息比會越多，而最好的情況是當 n=4，R 大約 N/2 時，可省下約 1/3 的訊息量。

revoked_num	saved message rate	my_3_saved_rate	my_4_saved_rate
1000	0.07	0.08	0.1
1501	0.11	0.11	0.15
2002	0.14	0.15	0.2
2503	0.17	0.18	0.23
3004	0.19	0.21	0.26
3505	0.21	0.24	0.28
4006	0.23	0.26	0.29
4507	0.24	0.27	0.3

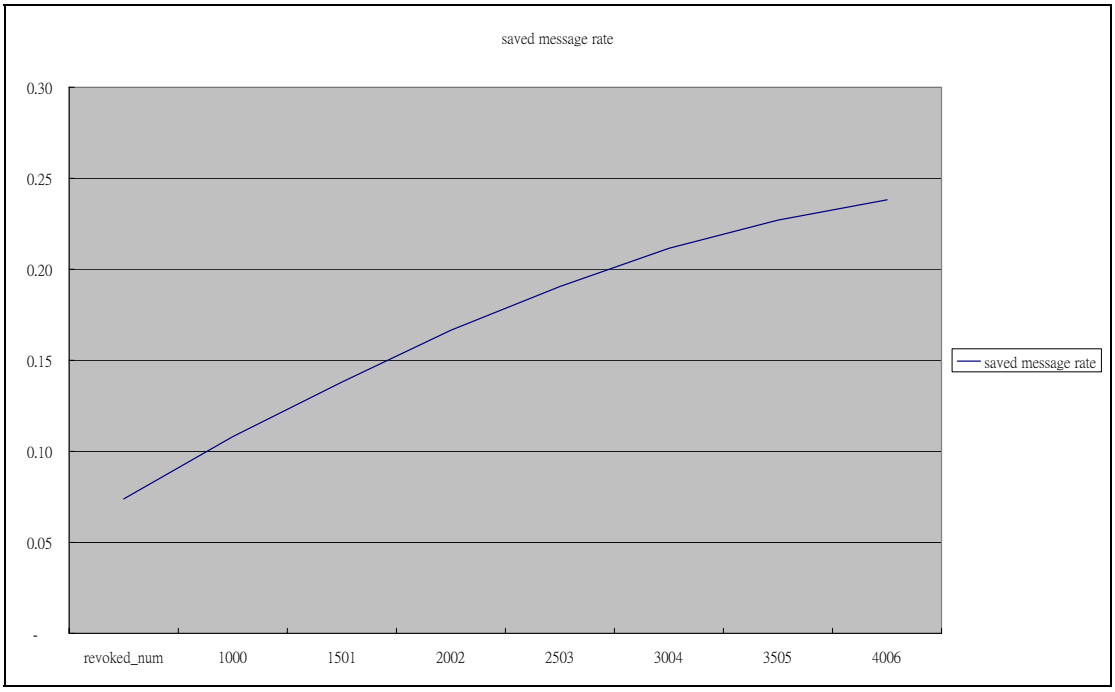
圖表 18. N=1000，T=400 時，基本型與 n=3，n=4 之一般型保留模式所節省下的訊息效率

下面的圖表 19，說明子集差別演算法跟基本型演算法所需的平均訊息量，說明我們所提出的演算法，確實有較子集差別來的有效率。



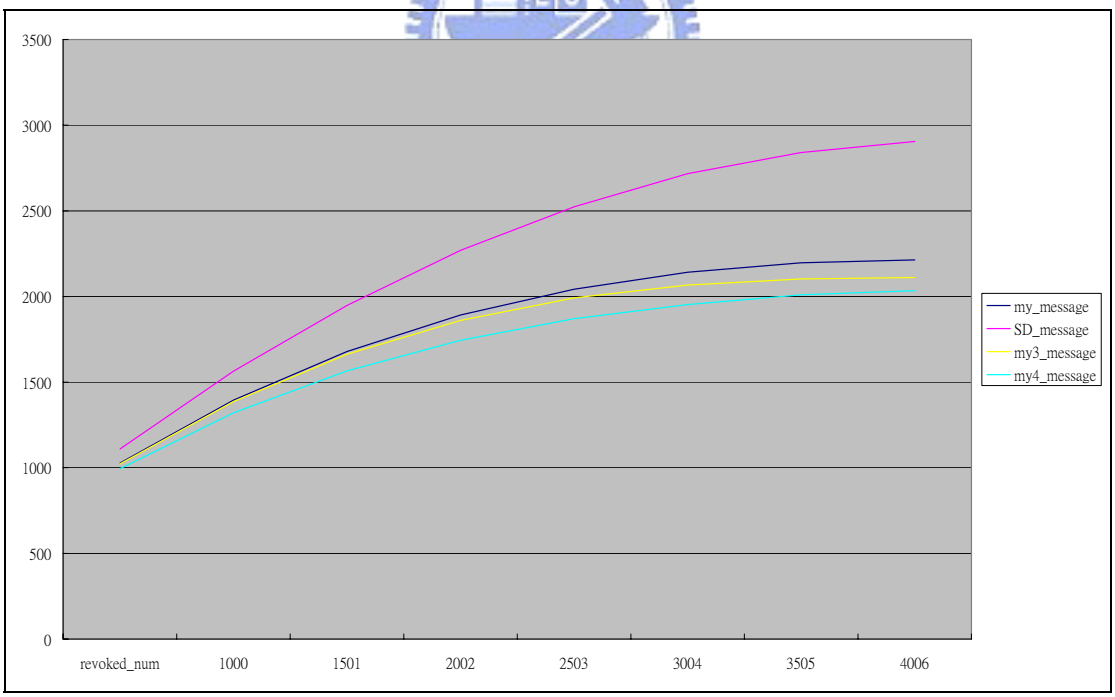
圖表 19.基本型與子集差別方法在不同註銷使用者數量時，所需要的訊息效率

接下來的圖表 20，是不同的 R 時，基本演算法所能夠省下的訊息跟 SD 演算法訊息量的百分比，說明出隨著被註銷者數量增加，所能夠節省的效率會越高。



圖表 20.基本型所可以節省的訓息效率曲線

接下來的圖表 21，是在不同 R 時，不同 n 的演算法所能夠節省下的訊息量，從圖中我們可以看出，最基本的 n=2 的形式已經可以省下大部分的訊息。



圖表 21.四個方法在不同數量的註銷使用者時所需要的訊息量

4.7 安全性討論

與一般的廣播加密演算法相似的，我們所提出的演算法的安全性，是基於管理中心所選擇的加密演算法而定，演算法本身只是做加密金鑰的管理與應用，理論上應該有與所用的加密演算法相同的安全程度，也就是說，假如存在有任何演算法 A，可以在 N 的多項式時間內，破解我們的廣播加密機制的話，我們就可以利用此演算法，造出新的演算法 B，可以在 N 的多項式時間內，破解底層所使用的加密演算法，而這樣的安全性，跟子集差別方法所能夠達到的安全性，是一樣程度的。儘管如此，最近有人提出[7]，類似子集差別演算法這類的無狀態性的廣播加密機制，因爲一開始分配好的加密金鑰就會被管理中心不斷的重複使用，而且一次的更新金鑰動作中，就同時有很多份用不同加密金鑰來加密相同交談金鑰的密文，而在已經分配的眾多加密金鑰中，只要攻擊者有機會得到其中一把金鑰，等於整個系統的安全性就已經失去。基於這個特性，從一段時間觀察管理中心傳送出來的更新金鑰訊息，已經有比暴力法來的有效的方式，可以在儲存空間、計算時間、與計算所需記憶體間，做一個有效的平衡，計算出至少一把加密金鑰出來，而只要一把加密金鑰不安全，有相當的可能性就代表此系統已經是不安全的。目前提出的攻擊方式，雖然較暴力法來的有效，所需的計算複雜度仍然需要次方式的時間，也許還可以有更短時間的攻擊方式，這是類似子集差別演算法等無狀態性廣播加密演算法都可能遭遇的攻擊，我們的演算法也同樣有這種風險。

第五章 結論與未來工作

到此為止，我們介紹完了廣播加密機制相關的方法以及本篇論文所提出的改良方式，在這一章，我們對整篇論文所提作一個結論，並提出未來的相關工作。

5.1 結論

廣播加密機制是一種在任何時間，保持只有合法使用者可以解開加密密文的機制。整個機制最主要兩個部份，第一個部分是把所有使用者分配到一個特定架構上，然後在此架構，定義出屬於此方法的子集合，通常，管理中心會分配給每一個子集合一個屬於此集合的金鑰，並且將此金鑰分配給所有屬於這個子集合的使用者；第二個部份是一個尋找覆蓋的方法，就是在已知的註銷使用者分配情況下，從定義的子集合中，找出可以覆蓋住所有合法使用者，並避免覆蓋到任何非法使用者的覆蓋集合來，利用這些覆蓋集合的金鑰來加密資料加密金鑰。

廣播加密機制從是否更新儲存裝置，可分為無狀態型機制，跟有狀態型機制，有狀態型機制需要較少的金鑰更新訊息的複雜度，但因為必須全程參與所有金鑰更新訊息，否則會無法解開之後的訊息，近幾年來大部分的論文發表，都是無狀態型機制，特別是從子集差別方法所延伸的演算法。

一個廣播加密機制所需要做的，就是找出適當的子集合定義以及在此子集合下找尋覆蓋的方法。尤其是定義子集合的方法，幾乎決定一個方法的好壞效率。近年來的所發表的方法多由子集差別方法所延伸，因此，定義子集合的方式也就大同小異。但是，我們深信研究的重心仍應朝著找尋更有效的子集合定義方式進行。

這篇論文，仍然沒有逃出子集差別方法的原理，但是，我們提出一個概念。並不是所有的非法使用者都一定要馬上被註銷其權限，這是在有狀態型機制中，早就有的概念，這種概念，在長時間持續性廣播，且廣播的加密並不是一定不能洩露的應用上，可以做更有效率的轉換。在這樣地應用上，只要大部分的時間，

非法使用者無法解開訊息，就可以得到廣播加密機制該有的管理，例如，付費節目。但是若所要廣播的資料是絕對不可以被解開的，例如軍事上的機密，就不適合採用這樣的概念。於這樣的概念，在無狀態型機制上，更是適合，因爲在無狀態型機制中，每一個非法使用者都會被考慮到，而不會因爲之前的金鑰更新訊息中，沒有把某位非法者的權限註銷掉，導致疏忽或者其他會引響到未來系統之事件發生的可能。

基於不必註銷掉所有的非法使用者，在提出新的路由器能力，幫助我們的方法情況下，在本篇論文中，我們提出一個方法，節省 SD 演算法中較無效率的訊息，而得到約 30%的改善。我們的演算法是針對 SD 系列演算法中特殊的訊息作處理，因此，其他與 SD 類似的演算法，例如 LSD 也可以應用類似方法取得改善。

5.2 未來工作

在今年 Eurocrypt(2005)會議上，Nam-su Jho 等人提出了一篇比 SD 來的有效率的廣播加密機制[9]，這篇方法所定義的子集合較子集差別方法來的複雜些，可以根據實際應用所需，調節與儲存複雜度、訊息複雜度有關的相關參數，在大部分的情況中，都較子集差別方法有效率。我們認爲在此廣播加密方法中，應該也會有類似 SD 方法中的較無效率的訊息，一樣會可套用我們所提出的概念，得到改善。

近年來大部分的相關方法，較著重在於找尋適當的子集合定義方式，對於在找尋覆蓋的方法上，大多是採取貪婪式的演算法，這樣的計算方式，對管理中心來說，是一種較爲減少負擔的方法，因爲，大多數學者認爲找尋一個絕對最小數量的覆蓋，有可能是一個 NP 問題。但是，從廣播加密機制被研究以來，尚未有論文對此觀點做一證明。

目前的廣播加密方法，都是採用一個集中式的管理：有一個管理中心，要負責管理底下眾多的使用者，註銷某些使用者後，還必須找出適當的加密金鑰。這樣的管理方式，對於管理中心來說是相當大的負擔，即使每個方法都有對其效

率做改善，仍然不適合用於數量龐大的系統。此外，大多數的應用上，中心除了做廣播的註銷等工作外，還得處理如應用上的政策等工作。在此情況下，我們相信更多的應用應該要採用分散式的管理方式，由地域或風格等因素劃分出多個分散式的管理中心，互相合作完成廣播加密的工作。研究一個分散式的廣播加密機制，相信也是一個必要的研究課題。

參考文獻

[1] D. Naor, M. Naor and J. Lotspiech, “Revocation and Tracing Schemes for Stateless Receivers,” in Proceedings of Advances in Cryptology – Crypto 01, Lecture Notes in Computer Science 2139, pp.41-62, 2001.

[2] C.K. Wong, M. Gouda and S.S. Lam, “Secure Group Communication using Key Graphs,” IEEE/ACM Transactions on Networking, Volume 8, pp.16-30, February 2000.

[3] D. Halevi and A. Shamir, “The LSD Broadcast Encryption Scheme,” in Proceedings of Advances in Cryptology –Crypto 02, Lecture Notes in Computer Science 2442, pp.47-60, 2002.

[4] M.T. Goodrich, J.Z. Sun and R. Tamassia, “Efficient Tree-Based Revocation in Groups of Low-State Devices,” in Proceedings of Advances in Cryptology – Crypto 04, Lecture Notes in Computer Science 3152, pp.511-527, 2004.

[5] M. Abdalla, Y. Shavitt, and A. Wool, “Key Management for Restricted Multicast Using Broadcast Encryption,” IEEE/ACM Transactions on Networking, Volume 8, pp.443-454, August 2000.

[6] T. Asano, “A Revocation Scheme with Minimal Storage at Receivers,” in Proceedings of Advances in Cryptology-Asiacrypt 02, Lecture Notes in Computer Science 2501, pp. 433-450, 2002

[7] M.J. Mihaljevic, M.P.C. Fossorier, and H.Imai, “Time-Data-Memory Trade-Off Based Cryptanalysis of Certain Broadcast Encryption Schemes,” from IACR Eprint Archive, http://eprint.iacr.org/2005/099 , 2005.

[8] S. Zhu, S. Setial and S. Jajodia, “Performance Optimizations for Group Key Management Schemes,” in Proceedings of the 23rd International Conference on Distributed Computing Systems, pp. 163--171, 2003.

[9] N.S. Jho, J.H. Cheon, M.H. Kim, and E.S. Yoo, “One-Way Chain Based Broadcast Encryption Schemes” in Proceedings of Advances in Cryptology-Eurocrypt 02, Lecture Notes in Computer Science 3494, pp. 559-574, 2005.