國立交通大學

資訊管理研究所

碩　士　論　文

無基礎行動網路環境上安全群播的密鑰管理機制

A Key Management Scheme for Secure Multicast on Mobile Ad Hoc Networks

研 究 生： 陳淑雯

指導教授： 羅濟群 博士

中華民國 九十四 年 六 月

無基礎行動網路環境上安全群播的密鑰管理機制

A Key Management Scheme for Secure Multicast on Mobile Ad Hoc Networks

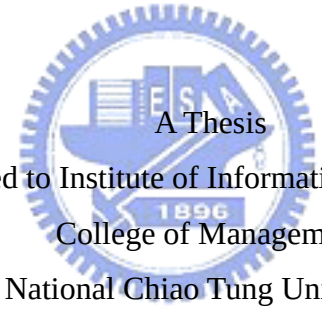
研 究 生： 陳淑雯　　Student: Shu-Wen Chen

指導教授： 羅濟群　　Advisor: Chi-Chun Lo

國立交通大學

資訊管理研究所

碩士論文



A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Business Administration

in

Information Management

June 2005

Hsinchu, Taiwan, the Republic of China

中華民國　九十四 年 六 月

無基礎行動網路環境上安全群播的密鑰管理機制

研究生：陳淑雯　　指導教授：羅濟群 教授

國立交通大學資訊管理研究所

摘要

隨著無線網路技術的進步，無線網路的應用亦趨普及。而在無線網路中，無基礎行動網路隨著技術之演進，及克服路由與服務品質等問題後，將使得它的應用範圍更廣。由於它不需藉由無線擷取器提供服務的特性，而是可以自我組成路由環境，亦因於此，它非常適用於災難救助、軍方作戰演訓及辦公室會議環境使用。但由於它具有動態拓樸及無線網路環境所形成的弱連結現象，使得它在實際應用面所遇到的安全問題較傳統的有線網路及無線網路環境更為複雜。

在網路環境中，可藉由點對點、點對多點或多點對多點方式完成資訊傳播，但由於無基礎行動網路的特性，點對點的方式較不適用，故群播在無基礎行動網路下的應用將劇增。而群播的安全性問題在此網路架構下完成群播之目的就顯得格外重要。另在叢集架構下之無基礎行動網路環境，路由及密鑰管理上較有效率。因此，本研究針對在叢集架構下的無基礎行動網路環境，就點對多點的安全群播問題做研究。

本研究提出在群播架構下就通訊點的單一節點/多個節點加入、離開及金匙管理問題提出一個安全機制與架構，以符合在無基礎行動網路中因動態拓樸與無線通訊本身之限制下，提供一個安全且具有效率的群播通訊環境。最後，於安全性分析上，本機制除滿足安全性需求外，另由於每個節點所握有的密鑰數量及因更新群組密鑰所需傳送的訊息量較其他機制少，故本研究所提的機制應可提供較佳的安全性與運作效率。

關鍵字：無基礎行動網路、安全群播、密鑰管理

A Key Management Scheme for Secure Multicast on Mobile Ad Hoc Networks

Student: Shu-Wen Chen Advisor: Dr. Chi-Chun Lo

Institute of Information Management

Nation Chiao Tung University

Abstract

Along with the technology evolution, and after overcomes the routing and quality of service problems, the Mobile Ad Hoc Networks, MANET, make the application in this environment more popular than before. MANET can self-organize routing and does not need access point, so it is useful for emergency operation, military environment and conference. Owing to dynamic topology and wireless environment its secure problems are more complex than wired or wireless network environments in the reality applications.

There are many ways for information dissemination during communication, like point-to-point, point-to-multipoint and multipoint-t-multipoint in the network environment; but due to the characteristic of MANET the point-to-point method is not applicable. Multicast is rapidly becoming a more application in the MANET environment. Therefore, the security problems of secure multicast are more significant, and focus this in our research. In addition, the cluster-based MANET can get more efficient than others in the aspect of routing and key management. We will propose secure scheme and scalable architecture for multicast while some node joins or leaves from the multicast tree and key management problems in the above architecture. We make multicast more efficient and secure under the restriction of the

dynamic topology and infrastructureless and wireless network constraint in MANET environment. Finally, security analysis present the schemes we proposed are more secure and efficient than the others.

Keywords: Mobile Ad-Hoc Networks, Secure Multicast, Key Management

誌謝

研究所的學習生活說長不長，說短不短，但隨著論文的完成，也代表研究所的學習生活即將告一個段落，對於這兩年來的時光，內心仍舊有些許的不捨。這二年來，不管在論文的指導上或生活、課業等其它方面，要感謝的人實在很多，其中最感謝的，是我的指導老師羅濟群老師。課業上，羅老師開授的課程讓我對網路及密碼學的領域有了深入的瞭解； Meeting 時所給予的看法與意見，讓我們能有正確的思考方向與啟發；每學期由老師帶隊的戶外之旅，讓我們享受到大自然的洗禮，培養了健康的身心，真的很感謝羅老師在這二年對我的指導。

另外一個對我課業及論文寫作上亦協助相當大的，則是博班俊傑學長，謝謝他在國科會計劃及論文上給我許多的建議跟指導，還有碩士班的秋儀同學，謝謝她那麼努力聽我論文的內容，協助我找出論文的盲點。碩二的同學們，謝謝大家在遇到困難時可以互相扶持、互相勉勵，還有碩一的學弟們，每週的讀書會報告都相當的精彩；當然，我還要特別感謝我的家人，尤其是我的父母，謝謝你們這麼的照顧我，讓我能在交大完成研究所學業，真的謝謝你們你們的全力支持。

目次

圖目次

表目次

一・緒論

在本章裡，主要說明本論文的研究動機、研究目標、研究方法及後續各章的簡單介紹。

1.1 研究動機

隨著無線網路技術的進步及設備和部署成本越來越低，在無線網路上的應用也越趨成熟，各式的通訊服務，如語音、文件、影像得以在通訊平台上傳輸資訊。目前無線網路分為兩類，需要無線擷取器(Access Point, AP)構成的無線區域網路，稱為基礎式網路(infrastructure network)，以及不需要無線擷取器，僅靠行動設備建立起來的無線網路稱為無基礎式行動網路(Ad Hoc Network)。

由於在有基礎架構之通訊環境須倚賴無線擷取器的存在方能完成資訊的轉送，因此，若有無線擷取器損毀、或因自然災害或因無線擷取器無法涵蓋的範圍，則此通訊管道失去作用，將造成節點間無法通訊。而無基礎架構之無基礎行動網路，正可解決上述問題[19] 。

因為無基礎網路架設容易，因此它非常適用於災難救助、軍方作戰演訓及辦公室會議環境使用。但它具有動態拓樸及無線網路環境所形成的弱連結現象，使得它在實際應用面所遇到的安全問題較傳統的有線網路及無線網路環境更為複雜，因此無基礎式網路上群組通訊的安全議題逐漸受到重視。

我們可以使用傳統的 IP 群播交換訊息，然而 IP 群播無法提供任何機制，來預防非群組成員存取群組交換訊息。雖然加密可用來保護群組交換的資料，但我們如何在無基礎行動網路上，安全並有效率的分散群組密鑰就是一個重要的議題。

在無基礎行動網路上架構群組通訊的環境，必須管理大量的群組成員，並面對成員經常的加入與離開。因此我們需要一個有效率的群組密鑰管理方法。許多

研究就此問題已提出不同的群組密鑰管理方法，這些方法可分為三部分：集中式的群組密鑰管理協定、半集中式的架構、以及分散式的密鑰管理協定。一般來說，在無基礎行動網路下採用分散式的架構較為適合，因為若此架構非分散式架構，就需要確保一個管理中心永遠存在無基礎行動網路中，且要負責計算群組密鑰，這樣對無基礎行動網路中的任一個成員都是不公平的。

然而，分散式密鑰管理協定卻需要先建構出類似樹狀的架構，而後進行密鑰及群播資料傳送，這在無基礎行動網路上不易達成，因此造成在實作上有困難度，所以本論文主要採用半集中式的架構，即以叢集架構之無基礎行動網路環境為基礎，建構出適當的群播環境，設計更安全且具效率的群組密鑰管理方法，並需兼顧每位群組成員對密鑰貢獻與掌握的公平性。

1.2 研究目標

本論文的研究目標是希望以叢集架構之無基礎行動網路環境為基礎，提出一個安全、有效率的密鑰管理協定(key management protocol)。本論文所提密鑰協定應用的群組通訊環境，是採用無基礎行動網路(Mobile Ad Hoc Network)，而非一般的實體有線網路或具基礎架構(infrastructure)的無線網路，目的是希望此空間內的成員能夠利用密鑰管理協定安全地、有效率地產生群組密鑰，以供後續群組通訊時的資料加密使用。

要達成安全群播，必須滿足以下四個條件[17]：

- 私密性(Confidentiality): 非群組成員，不能得知群組成員間傳送的資料。
- 身份認證(Authentication): 成員間資料的傳送必須提供有來源端的認證機制，以確認群組成員的身份。
- 向前安全性(Forward Security): 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。
- 向後安全性(Backward Security): 新加入的成員，不能得知他加入群組

前，群組成員間資料傳送的內容。

本論文旨在提出安全且具有效率的密鑰管理機制，以達成上述的私密性、向前安全性、以及向後安全性，至於身份認證則不在本論文討論範圍。且因無基礎行動網路環境下，群組成員可能因為訊號的變化，經常加入與離開，因此又提出在成員加入/離開時，如何進行群組密鑰更新的演算法，有效率的處理群組密鑰更新，並透過密鑰管理的機制，進行群播資料的傳送，達成安全群播。

1.3 研究方法

為能有效達成前述之研究目的，本研究乃採行下列研究步驟，首先進行文獻回顧整理, 瞭解目前群組密鑰管理方法，探討應用於動態群組群播的相關架構，深入研究各種密鑰管理方法的可行性與績效分析，歸納出應用於無基礎行動網路可能產生的問題，結合目前網路密鑰管理所制定的標準，發展有效的動態群組群播密鑰管理機制。.

研究流程如圖 1-1：

(1) 文獻收集

深入瞭解密鑰管理領域的研究主題，與最新研究狀況。

(2) 確定研究主題與架構

藉由相關文獻的整理與收集，瞭解目前正在進行的動態群組群播機制，針對無基礎式網路研究進行分析。

(3) 分析研究資料並對之前的方法進行問題分析

對於蒐集的文獻做一完整的分析及研究。

(4) 提出可支援多人加入/離開的密鑰管理方法並進行驗證、分析、與比較

利用之前所做的分析比較，規劃可實行的演算法，透過數學證明，驗證提出方法的優異性，並於數學證明中，設計多種案例進行比較。

(5) 歸納與未來研究方向

將數學證明所得的紀錄資料進行比對，驗證研究成果是否符合預期，提升密鑰管理機制的效率。歸納研究時遇到的問題，作為未來研究的方向。

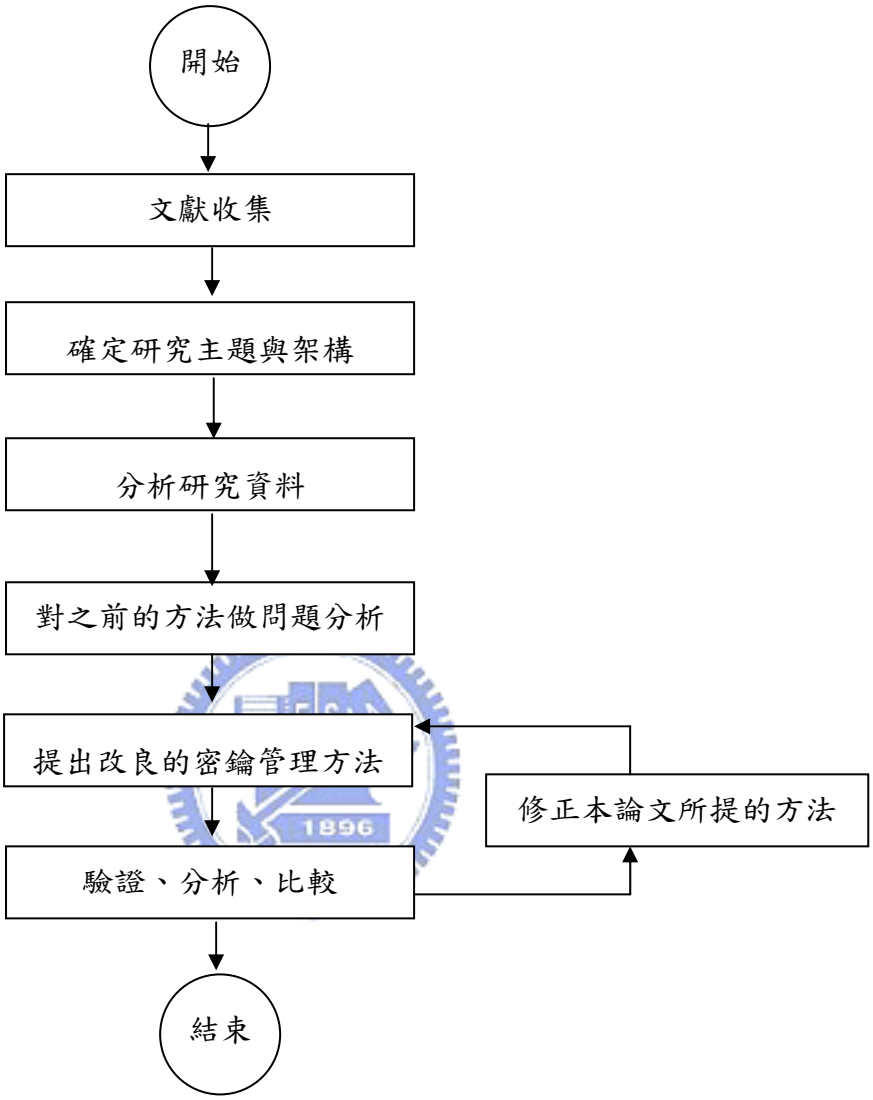


圖 1-1：研究方法流程圖

1.4 章節介紹

在第二章中，針對與本論文主題相關的研究做文獻探討，包括：無基礎行動網路的架構及安全上的挑戰、安全群播特性與群播密鑰管理方法的分類，並簡介一般以二元樹為基礎的密鑰管理架構以及 L. Morales 等人提出的 EBS 密鑰管理方法[18] 和二階式群播金匙管理方法[17] ；在第三章中，先對之前學者所提的方法做出問題分析後，接著提出改良的方法，並且透過本論文提出的密鑰管理方法及叢集架構的群播環境，設計群播資料傳送的方式；在第四章中，對本論文所提的方法做分析與比較，包括安全面及效率面；最後，在第五章中對本論文做結論及未來的研究方向。

二・文獻探討

在本章裡，主要介紹及說明與本論文主題相關的一些研究，包括無基礎行動網路的架構及安全上的挑戰、安全群播特性與群播密鑰管理方法的分類、並簡介一般以二元樹為基礎的密鑰管理架構以及 L. Morales 等人提出的 EBS 密鑰管理方法[18] 和二階式群播金匙管理方法[17] 等三大部份。

2.1 無基礎行動網路(Ad Hoc Network)

由於本論文所提出的密鑰管理協定主要是用於無基礎行動網路的環境，因此本節將對此網路做簡單及概念性的介紹，包括架構的介紹與在此環境上應注意的安全議題。

2.1.1 架構介紹

所謂的無基礎行動網路意指以對等方式進行無線網路存取，電腦之間直接做點對點無線連線，不需要透過無線基地台(Access Point, AP)。不過此模式的無線傳輸距離受限於電腦之間的距離，且各無線節點 (Node) 需設定相同頻道 (Channel)與 SSID 才能互相傳輸。

在無基礎行動網路環境中，節點間可以做直接的通訊，也能隨意移動，並繼續保持節點間連線的狀態。無基礎行動網路是由無線裝置自行建立的區域網路環境，其中並無無線擷取器或橋接器，它是一種能夠在沒有事先建置基礎架構的環境下，讓各個節點透過彼此點對點連結所臨時組成的網路，使得節點間能夠互相傳送資料，其架構圖如圖 2-1 所示。

無基礎行動網路特性即是節點間之通訊並不須預存無線基地台，它們彼此間能自我組成(Self-Organization)建構起通訊管道。但若兩個節點之前的距離超

過訊號接收範圍，則在此網路中的某一行動設備需藉由路由功能，搭起此兩節點的通訊通道。亦即具有動態拓樸的特性使得各個行動設備可以任意移動位置，且還能繼續和其他節點做溝通。因此，此架構除了具有傳統無線通訊的可移動性優點外，又因它具有無基礎架構的特性，其應用範圍更廣，例如，災難救助、軍方作戰演訓、辦公室會議環境及航空運用。

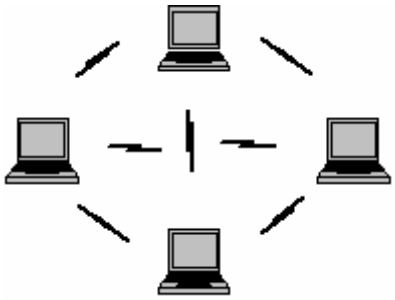


圖 2-1：無基礎行動網路架構

由於無基礎行動網路並沒有無線擷取器、路由器等裝置，因此，每個節點除了扮演一般的使用者之外，也必須同時具備有路由的能力。由於在無基礎行動網路環境，每個節點具有高度的動態特性，所以傳統的路由架構如 link-state 以及 distance-vector[1][2] ，均不適用。故以下即就無基礎行動網路路由架構做說明：

1. Proactive Protocol，也稱做 table-driven protocol：

   這個方式會經常性的去更新路由的路徑，當有封包需要被傳送時，路由的路徑通常是可以立刻得知的，使得封包能夠即時的被傳送出去。採用 proactive 方式的協定有：Destination-Sequenced Distance Vector protocol (DSDV)[3] 、Temporally-Ordered Routing protocol (TORA)[4] 、Lightweight Mobile Routing protocol (LMR)[5] 等。Proactive 協定的好處是，當有封包需要傳送時，路由路徑能夠很快決定；但是，單純的採用 proactive 方式並不能完全適用於無基礎行動網路上，因為節點會經常性的改變位置，維持路由的資訊會佔用網路很大的資源，且可能會有某些路由資訊在過期之前，都不會被用到，而浪費了網路的資源。

2. Reactive Protocol，也稱做 on-demand protocol：

這個方式只有在當有路由需求時，才會去進行路由路徑的決定。採用 reactive 方式的協定有：Ad hoc On Demand Distance Vector protocol (AODV)[6] 、Dynamic Source Routing protocol (DSR)[7] 等。由於在收到路由需求的訊息時，路由資訊可能不存在，因此，決定路由路徑將會造成封包傳送延遲時間較長，且在決定路由路徑時所需要的廣播訊息將會佔用很多頻寬，因此，單純的採用 reactive 協定也無法完全適用於無基礎行動網路上。

基於以上兩種路由協定，進而有學者提出應用於以叢集為基礎的無基礎行動網路(Cluster Based Mobile Ad-Hoc Networks)的路由協定：Cluster Based Routing Protocol (CBRP)[8] ，在這樣的架構下，由於叢集頭(cluster head)已經知道屬於它的叢集成員(cluster member)的資訊，因此，路由需要訊息的廣播只需要在叢集頭之間進行，不但能夠有效減少傳送路由訊息所需佔用的頻寬，也能較快速的決定路由路徑，這也是本研究最後選擇採用叢集式架構的原因之一。

2.1.2 安全上的挑戰

由於一般無線網路與有線實體網路傳輸媒介的差異，加上無基礎行動網路本身具有的特性，並非所有的安全協定皆適用於無基礎行動網路環境；因此，想要在無基礎行動網路上建構安全的通訊環境必須特別注意以下可能面臨到的挑戰[9] 。

1. 網路上面臨的攻擊：

由於無線網路上資料傳輸的媒介是空氣而不是實體線路，使得此環境下網路攻擊者的攻擊行為比在實體網路上更容易進行，也因為如此，使得無線網路的安全議題一直受人所重視，一般網路的攻擊行為可以分為消極攻擊與積極攻擊兩大類，消極攻擊指的是竊取網路上傳送的資料，而積極攻擊則不但竊取資料並竄改送出，可看出積極攻擊更具有威脅性。

2. 分散式或集中式架構：

由於無基礎行動網路是動態拓樸的架構，構成網路的每個成員都可以隨意移動，為了達到高度的存活性(survivability)及避免單一弱點的攻擊，應該採用分散式的架構，避免一個集中式的裝置，否則當單一集中式的裝置失敗後，整個傳輸網路就無法使用。

3. 省電及運算以及記憶體儲存空間的問題：

由於構成無基礎行動網路的節點都可以不受線路束縛的隨意移動，該設備一般都是靠電池供電，除了 Notebook 之外，還有 PDA 等，電力及運算能力以及記憶體儲存空間都有一定的限制。因此，在此環境下所採用的運算若是太過複雜或是需要儲存的資料過多，則可能造成設備執行運算及儲存上的負擔，並加速電力的耗損。

基於以上的討論，在無基礎行動網路環境中，節點之間要如何傳送訊息、溝通，是一個重要的議題。因此，有許多學者提出應用於無基礎行動網路的架構[10][11]，大致包含以下三種：

1. 集中式架構(centralized)：

由一個單一的單一管理者負責驗證成員合法性與安全傳送訊息。這樣的方式很明顯的在無基礎行動網路中是不容易達成的，因為我們無法確切知道網路的大小；除此之外，單一管理者萬一發生錯誤，將無法補救，且因為單一管理者是單一的點，很容易會遭受非法使用者的惡意攻擊，造成整個網路無法運作。

2. 階層式架構(hierarchical)：

將整個網路分成數個子網路，組合成一個階層式的架構，每個子網路也有各自的子網路，分別形成一樹狀的架構，並互相合作來提供安全的服務，這樣的架構能提供給大型的無線網路使用。但是階層式的架構主要仍然有下列三種問題：

(1)由於無線網路高度的動態性，經常需要改變路由的路徑，且節點也會移動，如何有效且快速的重新組織樹狀結構，是一個瓶頸；

(2)階層性的架構經常需要多階(multihop)的傳輸，使得傳送中的資料遭受竊取的機率更大；

(3)每一個行動機置都有可能遭受單一節點失敗的攻擊。

3. 叢集架構(cluster-based)[12] [13] ：

是將網路中的節點分成數個叢集，每個叢集中有各自的一個叢集頭及數個叢集成員，由叢集頭來負責成員的管理，及訊息轉送與傳送等動作。由於網路是被分成多個小叢集，因此管理網路成員將更方便；且成員的資料交換是由叢集頭協助傳送，不需透過多階傳輸，減少資料傳送過程中遭受攻擊的可能性。

叢集頭所需扮演的角色較複雜，因此需要有較強的計算能力，在動態的網路中，如何選出適合的叢集頭，並保證叢集頭不是一個非法的使用者，是無基礎行動網路的問題之一。叢集頭的選定方式大致有兩種[14] ：

1. Identifier-based Clustering：

依照使用者的身分編號來決定，通常可選定編號最低或最高的使用者來當作叢集頭，這樣的做法可較快速的選出叢集頭。

2. Connectivity-based Clustering：

依照使用者的連接性，有越多鄰居(neighbor node)的使用者將被選做叢集頭，如此做法可以使用較少的叢集數目就能夠包含整個網路。

至於要如何決定叢集的大小，即規定叢集成員與叢集頭之間最遠的距離，則必須考量到訊息傳送效率、叢集頭所需紀錄資訊的多寡等問題。假設同一叢集之中，叢集成員與叢集頭之間的距離越遠，則可以用少數幾個叢集就能夠包含整個網路成員，但是叢集頭所需紀錄它自己內部成員的資訊就要越多，且叢集頭要傳送給內部成員的訊息也必須經過較多 hop 數[12]，延遲訊息傳送的時間，也增加了訊息可能被竊取的機率；若叢集成員與叢集頭之間的距離越近，則叢集頭在跟內部成員交換訊息時，可以縮短訊息傳送的時間，也減少了訊息被竊取的可能性，但這樣的架構需要較多的叢集數才能包含整個網路的成員，且叢集頭也必須紀錄較多其他叢集的資訊。因此，要如何訂定叢集的大小，到目前為止有許多不同的叢集式架構提出，也有人提出除了叢集頭及叢集成員之外，應該有另一個角色：閘道點(gateway node)[12] ，用來負責連接兩個群組之間的通訊。

基於以上路由架構與安全性的考量，本研究將採用叢集式架構，來有效達到安全的訊息傳送以及成員管理的目的。

2.2 安全群播機制

本論文主要探討在無基礎行動網路下，如何達成安全群播。因此於本節中介紹安全群播特性，並簡介群播密鑰管理方法的分類。

2.2.1 安全群播特性

網路上資料的傳遞不外乎三種方式：

- 單播(Unicast)。單播就是一般所說的一對一傳輸，這也是傳統 Internet 的資料傳遞方式，資料只會送到指定的使用者端; 但是如果有三個使用者，則資料就必須傳遞三次，對於網路通訊頻寬會造成相當大的壓力。
- 廣播(Broadcast)則是一對多，但是並非網路上的所有成員皆需要收到傳送的訊息，因此此種方式會造成網路頻寬的浪費。
- 群播(Multicast)的技術就是希望能解決這個問題。資料接收端必須指定要加入網路上任一的群播群組(Multicast Group)，因此資料發送端只要送一次資料則不論資料接收端有多少，都能讓所有加入此群組的使用者接收到資料。

隨著網際網路與網路上通訊服務的快速成長，群組通訊變的越來越重要。群組通訊的服務包括：IP 電話、視訊會議、及協同式工作場所等等。同步、安全、及隱私對群組通訊是必要的。因此如何提供安全的群播，讓付費網路多媒體、責任性言論、公告及私人會議等群組應用，在傳送資料時能做到保密及認證的功能，是目前一項重要的研究課題。

所謂安全的群播，就是在群組成員在通訊時，必須提供溝通資料的私密性及可認證性的機制，滿足下列幾點的特性：[16]

1. 非群組成員，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需提供有來源端認證的機制。

3. 一個新加入的成員，不能夠得知他加入群組之前，群組成員間資料傳送的內容。

4. 一個離開群組的成員，不能得知其離開群組後，群組成員間資料傳送的內容。

要達到上述四點的安全群播特性，必需在群組成員間建立一個共享的加解密金匙，並作安全且有效率的管理。在[17] 一文中即對安全群播作詳盡的描述，以下摘其內容做說明：『一般安全群播協定需建立一群播金匙的管理機制(Key Management)，使得加解密金匙的共享不限定於兩者之間，而是由群組成員所共享，通常這把加解密金匙被稱為群組密鑰(Group Key，簡稱 GrpKey)』，另藉由輔助金匙達到群組密鑰更新的目的。

2.2.2 群播密鑰管理方法分類

一般而言，群組密鑰管理方法可分成三類[15] ：

◆ 集中式的密鑰管理(Centralized Group Key Management Protocols)

由單一個金鑰分佈中心或管理成員來產生群組金鑰。在集中式系統中，只有一個成員控制整個群組。集中式控制者不需依賴其他輔助成員執行存取控制和傳遞密鑰。然而，因為只有一個管理成員，當此管理成員被竊取或是發生重大失誤，整個群組密鑰管理系統就會失敗。也就是說，整個群組訊息的私密性決定於此管理成員是否能正常運作。當群組成員數目越來越多時，因為群組僅由單一成員管理，造成管理成員負擔過重且會失去效率。

◆ 半集中式的密鑰管理(Decentralized Architectures)

將整個群組分成多個子群組，並由各子群組管理者來產生密鑰。在半集中式的密鑰管理方法中，將整個大群組分割成數各小群組。每個小群組會有自己的小群組管理者。這個方法解決了將整個群組密鑰管理都交由單一管理者管理可能發生的負荷過重以及單一管理者失敗導致整個系統失敗的危機。但目

前此方法較少被採用，因為他仍須依賴一個集中的子群組管理者，負責控制子群組的存取，或是產生群組密鑰。這會使群組擴充的效率降低，因為他要靠集中式的子群組管理者和每個子群組管理者做接觸，以確定是否有新成員加入；且在產生密鑰的部分仍有風險存在，當集中式管理者無法產生群組密鑰，仍會造成群組通訊失敗。

◆ 分散式的密鑰管理(Distributed Key Management)

沒有任何群組控制者存在。群組密鑰可由所有成員共同貢獻部分資訊合作產生，或是由單一成員產生。然而交由單一成員產生密鑰是較不適切的作法，因為並非每個成員都具備產生密鑰的計算能力，例如亂數產生器。協同產生密鑰的處理時間和傳送的訊息次數會依群組成員線性增加。採用協同產生密鑰的方式時，每個群組成員都必須隨時注意群組關係的變動，例如新成員加入或舊成員離開。

由於本研究採叢集式的拓樸協定，因此於安全群播之密鑰管理協定將以半集中式的方式達到安全群播的效果。本論文之安全群播機制基於[18] 作者所提的 EBS (Exclusive Basis Systems) 並結合 sum of product 的觀念應用於叢集內與叢集間的密鑰管理與動態成員加入與離開，使其達到安全群播之效果。以下先介紹以傳統二元樹架構進行密鑰管理的方法，之後再對 EBS 機制做完整性介紹。

2.3 二元樹架構密鑰管理方法

在[20]這篇 RFC 規格書中，採用傳統的二元樹架構，管理群組密鑰。樹中的每個節點都代表一把鑰匙，根節點代表此次會議或是群組的通訊密鑰，加密群播資料送給群組成員；葉節點代表每位群組成員個別擁有的私有密鑰；而內部節點(除了根節點)則代表輔助金匙(Administrative Keys)，輔助金匙在成員離開時，能使系統快速的進行群組密鑰更新。在二元樹中，每個群組成員由對應他的葉節點到根節點，會有一條 Key Path，也就是說每個群組成員會知道其 Key Path

上的每把鑰匙。因此，如果二元樹是完全二元樹或是平衡二元樹，每個成員會握有 $(\log_2 n)+1$ 把鑰匙，等同於樹的高度，包含 1 把私有密鑰、1 把通訊密鑰以及 $(\log_2 n)-1$ 把輔助金匙，如圖 2-2 所示。

在圖 2-2 中，我們將每把鑰匙視為一個集合體，在此集合體中的元素，即表示握有此把鑰匙。舉例來說，編號為 00 的鑰匙，集合為{0,1}，表示成員 0 和成員 1 握有此把輔助金匙；而編號為 0 的鑰匙，集合為{0,1,2,3}，表示成員 0、成員 1、成員 2、成員 3 皆握有此把輔助金匙。另外，群組通訊密鑰 λ 的集合元素為{0,1,2,3,4,5,6,7}，表示每位成員皆握有群組通訊密鑰。

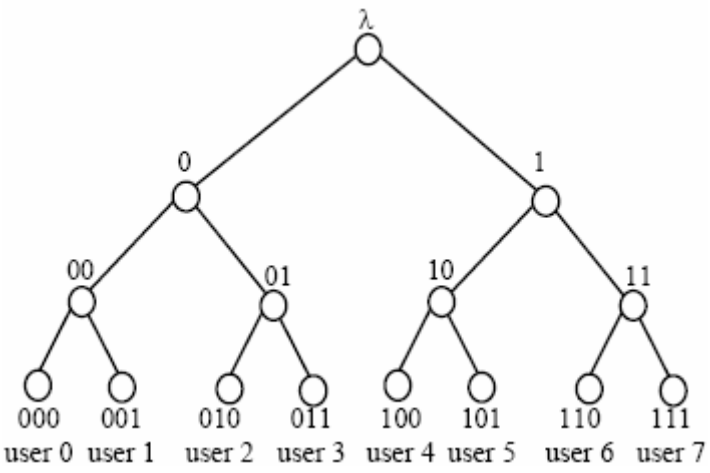


圖 2-2：以二元樹為基礎的密鑰管理架構[18]

當有成員想離開群組時，僅需送出三個更新群組密鑰的訊息即可。假設成員 0 想離開群體，我們可以用編號為 1 的輔助金匙加密新的群組密鑰後，送給所有成員，因為僅有成員 4、成員 5、成員 6、成員 7 握有此把輔助金匙，因此僅有此四位成員可解開此訊息取得新的群組密鑰。另外，用編號為 01 的輔助金匙加密新的群組密鑰與輔助金匙 0 後，送給所有成員，因為僅有成員 2、成員 3 握有此把輔助金匙，因此僅有此兩位成員可解開此訊息取得新的群組密鑰。最後，用編號為 001—user1 所握有的私有金鑰，加密新的群組密鑰與輔助金匙 0 和 00 後，送給所有成員，因為僅有成員 1 握有此把私有密鑰，因此僅有成員 1 可解開

此訊息取得新的群組密鑰。透過傳送此三個更新密鑰的訊息，便可更新而不讓離開的成員 0 得知新的群組密鑰。此即為一般二元樹架構下更新群組密鑰的作法。

2.4 EBS(Exclusive Basis Systems)群組密鑰管理方法

在[18] 這篇論文中，提及當群組的成員經常變動時，如何能有效率管理群播的群組密鑰的方法。此方法稱之為 EBS，它改善了目前以二元樹為基礎及其他相關系統的密鑰管理方法。

在群體中，每位成員不僅握有群體密鑰，還個別握有數把用來協助群體密鑰更新的輔助金匙。這篇論文提出一個技術，稱為 EBS，此為一個群體密鑰管理的組合公式，即如何在成員數為 n 下，求出最佳的 k 和 m。n 指的是群體成員數，k 則為每個成員握有的輔助金匙數，m 是每次更新群體密鑰所需送出的訊息數。在這篇論文中，描述了單一成員加入/離開的演算法，並且驗證了大型群體採用 EBS 進行密鑰管理的效率。

EBS 定義：令 n、k、m 皆為正整數，1 < k、m < n。我們將 EBS(n, k, m) 表示為數個子集合（子集合內容為[1, n] = {1, 2, …, n}）的集合體，令它為 Γ，即 $\Gamma=\{A_1, A_2, \cdots, A_i\}$。每個整數 t∈[1, n]滿足以下兩個特性：

(a) t 最多只能出現在 Γ 中的 k 個子集合。

(b) 在 Γ 中，當有 m 個子集合做聯集，即 $A_1 \cup A_2 \cup \cdots \cup A_m$，使得 $\bigcup_{i=1}^{m} A_i$ = [1, n]-{t}（表示若要排除掉 t 元素，需要 m 個在 Γ 中的子集合做聯集）。

舉例如下：EBS(8, 3, 2)是一個數個子集合的集合體 Γ = { A_1={5, 6, 7, 8}, A_2={2, 3, 4, 8}, A_3={1, 3, 4, 6, 7}, A_4={1, 2, 4, 5, 7}, A_5={1, 2, 3, 5, 6, 8}}。我們可以簡易的驗證每個 t∈[1, 8]在 Γ 所有子集合中僅出現 3 次，且每個成員都被在 Γ 中的兩個子集合聯集後排除。就像下面所示：

$$[1,\ 8]\ -\ \{1\}\ =\ A_1 \cup A_2$$
$$[1,\ 8]\ -\ \{2\}\ =\ A_1 \cup A_3$$
$$[1,\ 8]\ -\ \{3\}\ =\ A_1 \cup A_4$$
$$[1,\ 8]\ -\ \{4\}\ =\ A_1 \cup A_5$$
$$[1,\ 8]\ -\ \{5\}\ =\ A_2 \cup A_3$$
$$[1,\ 8]\ -\ \{6\}\ =\ A_2 \cup A_4$$
$$[1,\ 8]\ -\ \{7\}\ =\ A_2 \cup A_5$$
$$[1,\ 8]\ -\ \{8\}\ =\ A_3 \cup A_4$$

一個維度為(n, k, m)的 EBS 集合體 Γ，表示在群體中有編號為 1 到 n 的 n 個使用者，而密鑰伺服器（Key Server）保管集合體 Γ 中所有子集合的輔助金匙（即為 A_i）。若 A_i 出現在 Γ 中，表示出現在 A_i 此子集合中的所有成員都握有這把輔助金匙，像上例 EBS(8, 3, 2)中，僅有成員 5、6、7、8 才握有輔助金匙 A_1。而對於 $\bigcup_{i=1}^{m} A_i$ =[1, n]-{t}此特性，即表示當密鑰伺服器想驅逐某位成員時，可以使用 $\bigcup_{i=1}^{m} A_i$ 中所有的 A_i 來加密新的群體密鑰送給留下來的成員，換句話說，當有成員離開群體，密鑰伺服器僅需群播出 m 個用 $\bigcup_{i=1}^{m} A_i$ 中所有 A_i 加密過的訊息給所有群體成員，如此可確保除了 t 以外的所有成員都可獲得新的群體密鑰。

當有成員離開時，除了更新群體密鑰，該成員握有的輔助金匙亦需更新以避免串謀攻擊，此論文建議可採用雜湊函數 f(新群體密鑰, 舊 A_i)來算出新的 A_i，如此不但可確保僅原先握有該輔助金匙者才可得出新的 A_i，亦可避免由密鑰伺服器產生新 A_i 後，以舊 A_i 加密後再傳送給所有成員的流量浪費。

現在可以開始建構 EBS，首先我們先列舉如何在 k+m 個物件中，形成 k 個子集合的可能方法。對於所有的 k 和 m，我們令 Canonical(k, m)為 $\binom{k+m}{k}$ 以在 k+m 個物件中，形成 k 個子集合。舉例矩陣 A=$\binom{5}{3}$，如圖 2-3 所示：

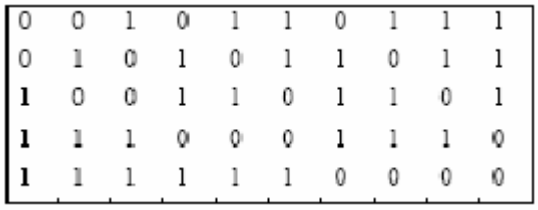


圖 2-3：子集合 A_i 與成員間關係[18]

此矩陣即可用來管理當群體成員為 10 時的輔助金匙分配表。表格中的每列表示輔助金匙 A_i 分配的成員（1 表示有被分配到 A_i ），每欄即為每位成員握有的 A_i（1 表示握有 A_i)。在此例中，n=10，k=3，m=2。因此當有某位成員要離開時，僅需以m=2 兩個 A_i 加密送出rekey訊息即可。假設成員 1 離開此群體，僅需以 A_1（握有 A_1 者包含M_3、M_5、M_6、M_8、M_9、M_{10}(M_i表示編號為i的成員)）和 A_2(握有 A_2 者包含M_2、M_4、M_6、M_7、M_9、M_{10})分別加密新群體密鑰後送出，則除了成員 1 以外的所有其他成員都能解開此訊息。需注意的是 $\binom{k+m}{k}$ 必須大於等於n，亦即每位成員握有的Key String都不相同，如此才可確保此密鑰管理系統的安全性。

當新成員加入一個群體大小為 n 的群體時，若 $\binom{k+m}{k}$ 仍大於或等於 n+1，則僅需分配新的 Key String 和對應的輔助金匙給該成員。反之，若 $\binom{k+m}{k}$ 小於 n+1，則可採用兩種方法：

1. 增加每個人握有的輔助金匙數：

   即除了新加入成員以外的其他所有成員，都取得新增加的輔助金匙 A_{k+m+1}。則密鑰伺服器將先前的 k+m 把輔助金匙（不含新產生的輔助金匙）任選 k+1 把分配給新加入成員。

2. 增加 rekey 訊息的送出次數：

   即除了新加入成員取得新增加的輔助金匙 A_{k+m+1}，其他所有成員握有的輔助金匙都維持不變。因此密鑰伺服器需將先前的 k+m 把輔助金匙（不含新產生的輔助金匙）任選 k-1 把分配給新加入成員。

當成員離開時，關於 rekey 的運作，即採用上述聯集所產生的輔助金匙，來

加密新的群體密鑰送出，確保離開的成員無法解開。但除了 rekey 的部分，我們仍須考量如何減少密鑰伺服器所管理的輔助金匙數，每位成員握有的輔助金匙數，以及 rekey 訊息送出的次數。

因此當新成員離開一個群體大小為 n 的群體時，若可以較少的 k 或 m 滿足 $\binom{k+m}{k}$ 仍大於或等於 n-1，則需減少每位成員握有的輔助金匙數或 rekey 訊息送出的次數，以達到 EBS 的最佳化。

若離開成員 y 即是最後加入者，只需將系統回復到 y 加入前的狀態即可（及回復到先前所採用的 k 及 m 個數）。但當離開成員和最後加入成員的對象不同時，則可採用以下運作方式：

當成員 x 要離開此群體，而成員 y 是最後加入者，則

(1)將 x 和 y 驅逐出此群體（x 和 y 握有的 A_i 都會在 rekey 的訊息中被更新）；

(2)增加成員 y 回此群體；

(3)但成員 y 會被分配先前成員 x 的 Key String 和新的對應 A_i。

EBS 在密鑰管理方法上提供一個新的架構，EBS 所需採用的輔助金匙個數和 rekey 訊息送出次數，都明顯優於以二元樹資料結構來管理輔助金匙。

由於本研究僅就密鑰管理與單人加入/離開問題做討論，並未說明如何有效做安全群播與多人加入/離開之機制研究，因此，本研究除了將 EBS 方法引至無基礎行動網路環境，並結合 sum of product 概念達到上述效果。以下簡介在二階式群播金匙管理方法[17] 中如何使用卡諾圖迅速找出 sum of product，以有效率達成群組密鑰管理。

2.5 二階式群播金匙管理方法

在二階式群播金匙管理中，將群組分成兩個區域：一是路由器區域(Router Domain，簡稱 RDomain)，另一是成員區域(Member Domain，簡稱 MDomain)，如

圖 2-4 所示。

每個群組只會存在一個路由器區域；路由器區域是由邊界路由器組成，該邊界路由器在群組進行後將被稱為子群組管理者(Subgroup Manager)。在眾多的子群組管理者將有一個會被特別稱為群組管理者(Group Manager)。群組管理者負責管理路由器區域內子群組管理者在加入及離開群組通訊時群播金匙的分配和更新等動作。

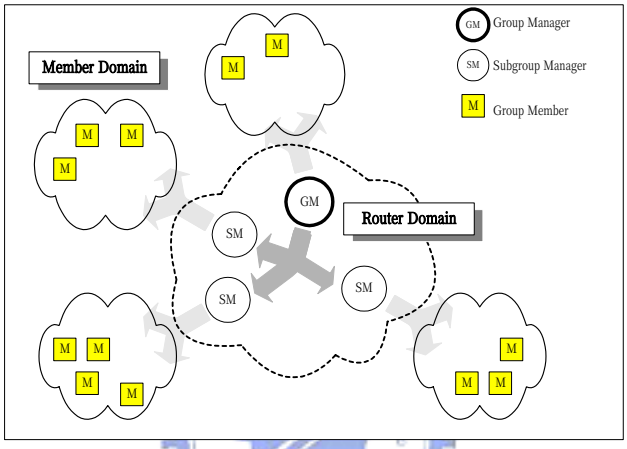


圖 2-4：二階式群播管理模型[17]

成員區域則是由一個子群組管理者及其管轄範圍內的已註冊群組成員所組成，子群組管理者將負責管理該成員區域內的成員加入及離開群組通訊時群播金匙的分配和更新等動作。

二階式群播金匙管理利用分層管理的觀念，將整個群組成員的管理負擔分散到各個子群組管理者上，讓群組管理者只負責管理子群組管理者，並不直接管理群組成員。利用這樣的階層化觀念，將使得安全群播群組的管理擁有可量化的特性。

基本是我們亦可稱其採用的是叢集架構，由群組管理者管理子群組管理者，另交由子群組管理管理者直接管理屬於他的群組成員，由於網路是被分成多個小叢集，因此管理網路成員將更方便。

每個成員區域內，會維持一個輔助金匙分散樹，此樹狀架構將由該區域的子群組管理者維護，子群組管理者依照該成員區域內的成員名單建立靜態的輔助金匙分散樹。其使用完全二元樹(Complete Binary Tree)的方法來建立成員區域內的輔助金匙分散樹，如圖 2-5 所示。

在圖 2-5 中，由子群組管理者依照成員建立一個完全二元樹的輔助金匙分散樹。圖中每個圓點代表一個輔助金匙節點，而每個方形成員節點則代表著一個群組成員。由於為完全二元樹的架構，會有部份成員節點是空的，表示沒有代表任何成員。

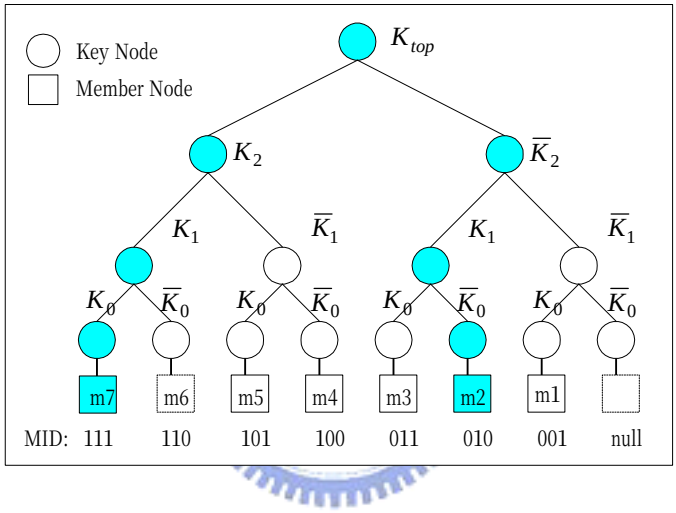


圖 2-5：用完全二元樹建立成員區域輔助金匙分散樹[17]

對於輔助金匙分散樹上的金匙配置，子群組管理者將先產生 $2^{\lceil \log N \rceil}+1$ 把的輔助金匙，其中 N 為該成員區域內的成員個數，然後配置到輔助金匙節點。樹根(root)節點將被配置 K_{top}，在樹的第 i 層每個節點其左邊兒子節點(left child)配置金匙 K_{i-1}，而右邊兒子節點(right child)則配置金匙 $\overline{K}_{i-1}$，$i>0$ 且 K_{i-1} 和 $\overline{K}_{i-1}$ 並沒有任何關係。

每個成員節點將配置一MID(Member ID)代表著該成員節點到樹根節點間所經過輔助金匙節點，同時也表示該註冊成員加入群組通訊時時被配置到的輔助金匙。MID用二元字串(Binary String) $b_{n-1}b_{n-2}\cdots b_0$來表示，其中n為二元字串的長度且$n=\lceil \log N \rceil$，而b_i可為 0 或者是 1，如圖 3 中m6 的MID為 110，其中b_2=1、b_1=1、

b_i=0。如果註冊成員MID的b_i=1 則表示註冊成員在加入群組通訊後會分配到K_i，反之b_i=0 即表示註冊成員會分配到$\overline{K}_i$，而每個成員皆會分配到K_{top}。如圖四中m6的MID為 110，即表示m6 在進入群組後將會取得輔助金匙K_2、K_1、$\overline{K}_0$及K_{top}，m2 的MID為 010，表示m2 在進入群組後將會取得輔助金匙$\overline{K}_2$、K_1、$\overline{K}_0$及K_{top}。

在處理多重成員離開群組通訊時，透過使用布林代數化簡的方法，將更新群組金匙所需的的訊息個數減到最少。例如在圖 2-5 中，若成員 m7 及 m2 同時要離開群組，並且 m6 並未加入群組，MID 000 並未配置。我們可以 MID 為輸入，並依下列原則建立成員真值表，如表 2-1 所示。

- 輸出為 0 表示該 MID 成員將離開群組通訊。
- 輸出為 1 表示該 MID 成員沒有要離開群組通訊。
- 輸出為 X 表示該 MID 成員並未加入群組通訊，或者該 MID 並未配置。

表 2-1：成員真值表

Input(MID) $b_2b_1b_0$	Output
000	X
001	1
010	0
011	1
100	1
101	1
110	X
111	0

(a)卡諾圖　　(b)卡諾圖化簡圈選

圖 2-6：多重成員離去之卡諾圖化簡[17]

由於 MID 的字元(bits)同時代表著成員所擁有的輔助金匙，所以我們可以利

用卡諾圖或是列表法來化簡成員真值表，並將化簡結果以積之和(sum of product expression)的方式來表示，而化簡結果即為非離去成員間所擁有的共同輔助金匙。如圖 2-6 所示使用卡諾圖方法化簡表 1，其化簡結果$\bar{b}_1 + \bar{b}_2 b_0$。在化簡結果中的每個積項(product term)即代表著部份非離去成員間所共有的一組輔助金匙，而全部的積項將能含蓋全部的非離去成員。例如$\bar{b}_1$代表輔助金匙$\overline{K}_1$，而$\overline{K}_1$被 m5、m4 及 m1 所共有，$\bar{b}_2 b_0$代表輔助金匙$\overline{K}_2$及K_0，並且只有 m3 和 m1 同時擁有$\overline{K}_2$及K_0，而 m5、m4、m3 及 m1 則是全部的非離去成員。

在求得共同輔助金匙組後，子群組管理者便可以利用各組的同共輔助金匙來對更新群播金匙的訊息進行加密。如果共同輔助金匙組的金匙數目為 1，則只要直接利用該輔助金匙來加密，而如果共同輔助金匙組的金匙數目超過 1，則使用複合金匙的方式將多把輔助金匙結合成一把，並利用該複合金匙來加密。在本例中，最後子群組管理者必需分別使用輔助金匙$\overline{K}_1$及複合金匙$K_{\overline{20}}$($K_{\overline{20}} = f(\overline{K}_2, K_0)$)來加密群組金匙更新訊息。

本論文為了達成成員加入/離開亦能有效率的更新密鑰，故在叢集架構下亦採用卡諾圖進行 sum of product 的化簡，在群組成員加入及離去群組的動作非常的頻繁或是群組成員分散情形極端的時候，叢集架構下採用卡諾圖進行化簡將能提供更好的群組金匙更新效率。

三、支援多人加入/離開之群組密鑰管理協定

在第一章的研究動機中，本論文提出想達成的目標及方向，因此收集、研究了許多與本論文相關的文獻，如第二章所介紹。在文獻探討的過程中，了解無基礎行動網路的特性及安全上的挑戰、了解為什在無基礎行動網路環境上較適用以叢集式為基礎的群組密鑰管理協定，而不採用分散式的密鑰協同協定，也了解利用 EBS 和卡諾圖進行密鑰管理的好處，接著本論文提出一個能支援多人同時加入/離開的密鑰管理協定，在此章中，先對問題做分析、定義，接著再對本論文提出的方法做詳細的介紹。

3.1 問題分析

由於無基礎行動網路應用於無線網路環境上的設備，其電力、運算能力以及儲存空間常被假設有一定的限制，綜觀多種架構下的群組密鑰管理協定，發現採用 EBS 機制進行密鑰管理，可以大幅減少每個成員的運算時間、儲存空間以及需送出的 Rekey 訊息，即減少每位成員的運算及儲存負擔，並可降低 Rekey 訊息佔用的頻寬，因此本論文所提的方法即基於以 EBS 機制結合卡諾圖構成安全群播的密鑰管理機制。

此外，無線網路的封包較一般實體網路更容易發生群組成員隨時加入/離開，因此不管單人加入/離開或是多人同時加入/離開，如何快速安全並有效率的更新群組密鑰，是本論文想要解決的問題，另外，當有一個完善的密鑰管理機制，如何在叢集式架構下，安全並迅速的將資料送到目的地，也是本論文欲探討的內容。

在仔細檢視於第二章所述的 EBS 機制後，發現此方法僅提出以 Bit String 來進行密鑰更新，卻未深入提及在無基礎行動網路上，應極易發生的多人加入/離開情況下，如何做群組密鑰的更新。因此在本論文中，透過 EBS 機制與卡諾圖的搭配，設計了如何進行多人加入/離開的群組運作與密鑰更新機制，且討論如何在這樣的機制下進行安全群播。

因此基於[19]此篇論文中提出的 EBS 方法，本論文欲延伸並改進的部分有：

- 結合卡諾圖協助群組訊息傳送

  將每位群組成員握有的輔助金匙 Bit String，透過卡諾圖進行運算，可有效率地找出可用來加密訊息的輔助金匙，避免群組訊息被群組外的其他人或是不應取得此訊息的群組內成員得知。

- 延伸 EBS 機制，設計可支援多人同時加入/離開演算法

  本論文延伸 EBS 機制下的單人加入/離開演算法，設計可支援多人加入/離開演算法，滿足原先 EBS 機制的優點：即每個群體的輔助金匙及需送出的 Rekey 訊息都達到最小值。且透過卡諾圖的運算，在多人加入/離開演算法的實作上能更有效率傳送 Rekey 訊息。

- 叢集架構下的安全群播

  深入探討如何藉由本論文所提出的密鑰管理機制，在叢集架構下安全的在叢集間及叢集內傳送群組訊息，以達成安全群播的目標。

3.2 支援多人加入/離開之密鑰管理協定

基於前一節裡的問題分析描述，本論文所要做的方向，即基於[19]作者所提的機制完成群組密鑰傳送，但以最少的輔助金匙協助群組密鑰的更新；並解決[19]更新群組密鑰的運作效率問題；除了改良群組內多人加入/離開如何更新群組密鑰的演算法之外，還增加了如何透過卡諾圖運算，進行群組密鑰及資料的加密與傳送，讓安全群播得以達成，以下即對本論文所提的密鑰管理協定做詳細的介紹。

3.2.1 成員合作的傳輸架構

本研究乃基於叢集架構下討論無基礎行動網路的安全群播機制。因此，它包括了叢集內與叢集間動態環境下的密鑰管理機制，以滿足安全群播之目標。而之所以採用叢集架構，其原因為它具有較佳的路由效率。因此，在本研究中我們假設能和叢集頭直接進行通訊(Single Hop)的所有成員形成一個叢集，在叢集內的所有成員共同分享一把子叢集密鑰。不同叢集間亦可透過叢集頭，使用 Inter-cluster key 互相通訊。而任何一個節點均可加入或離開某一叢集。

在本研究機制中是以編號最小者做為該叢集之叢集頭，且此叢集頭具有可信賴的特性，除非此叢集頭離開了此叢集，則必須重新找叢集頭，並重建此叢集內所有成員的信賴關係，且該叢集頭必須重新取得叢集間的群組通訊密鑰及輔助金匙。於叢集間亦以編號最小的叢集之叢集頭當做此群組之叢集頭。同樣的，它必須具備與叢集內之叢集頭同樣的特性，它的目的是為了實現跨叢集的安全群播。

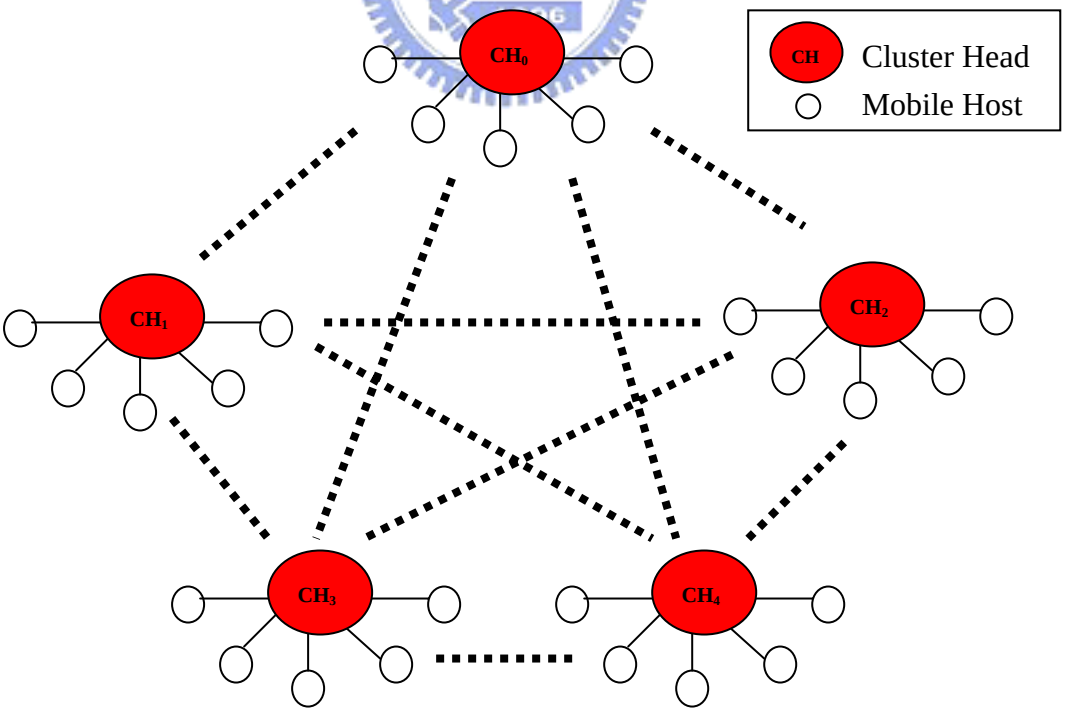


圖 3-1：叢集架構之無基礎行動網路

圖 3-1 即為本研究之叢集架構圖，以CH標示者為該叢集的叢集頭，編號最小的叢集頭(CH_0)為群組控制者，即此群組之叢集頭。

另先就本研究所須之假設在此做描述：

假設 1：

此群組通訊環境共區分成n個叢集，即 $GS = \{Cluster_i \mid \forall_i = 0,...,n-1\}$ ；每個叢集擁有m個成員(節點)，即 $Cluster_i = \{M_{i,j} \mid j = 0,1,2,...\}$ ，其中每個叢集內的成員個數可以不一樣， M_{i0} 即為 $Cluster_i$ 的叢集頭，標示為CH_i。

假設 2：

$M_{0,0}$ 為此群組通訊主要控制者，由它來產生叢集間的群組密鑰及各叢集頭的輔助金匙，做為叢集間更換群組密鑰之用外，另它亦扮演此叢集內之叢集頭角色；而 $M_{i,0}$ 為第 *i+1* 個叢集之叢集頭，其目的是為了產生此叢集之子叢集密鑰及該叢集內所有成員的輔助金匙，做為叢集內更換子叢集密鑰之用。另所有叢集之叢集頭均為可信賴的節點，由它們來協助完成密鑰管理與群播工作。

假設 3：

$M_{0,0}$ 與 $M_{i,0}; \forall i = 1,2,...,n-1$ 間已存在一個秘密金鑰 PW_i ；同理每一個叢集頭 $M_{i,0}$ 與該叢集內的所有成員 $M_{i,j}; j = 1,2,...$ ，亦存在一個秘密金鑰 SPW_j 。

假設 4：

叢集內之叢集頭 $M_{i,0}$ 或此通訊群組之叢集頭 $M_{0,0}$ 必須協助合法的群組成員完成群播事宜。

3.2.2 密鑰管理機制

關於在本論文中所提出之密鑰管理演算法中，使用的相關符號定義，皆說明如表 3-1：

表 3-1：本論文密鑰管理方法中的符號定義

符號	說明
CH_i	群組通訊環境中的編號為 i 的叢集頭
M_{ij}	群組通訊成員，編號為 i 之叢集中，編號為 j 的成員
PW_i	群組通訊主要控制者握有的其他叢集頭之私密金鑰
SPW_j	各叢集頭握有的該叢集內所有成員之私密金鑰
A_k	群組通訊環境中，叢集間的輔助金匙
A_{ik}	各叢集內所使用的輔助金匙
GK	群組通訊環境中的群體密鑰，即 inter-group key
Sub-GK$_i$	各叢集內使用的子叢集密鑰，即 intra-group key
E_{Key} {data}	使用對稱式加密將群組通訊資料“data”以密鑰“Key”加密
$f(\ )$	單向雜湊函式

由於群播具有點對多點遞送之效果，故非常適用於無基礎行動網路環境。本小節就群播密鑰之管理，做詳細介紹。

本機制採[18] 所提之 EBS 機制，並使用 sum of product 的機制以達快速化簡之目的，如此使得群播所需之輔助金匙及當有成員離開群組，完成更新此群組密鑰之後，將新的群組密鑰送至現有成員所需使用之輔助金匙能很快的計算出來。於本研究中，安全群播之密鑰管理，亦區分叢集間與叢集內之密鑰管理機制。亦即是說，對某一叢集而言，該叢集頭必須產生子叢集密鑰鑰及協助叢集內成員產生子叢集輔助金匙，作為該叢集內任一節點欲進行叢集內群播之用；另對叢集間而言，$M_{0,0}$ 必須協助剛加入此通訊群組之叢集代表，即是該叢集之叢集頭 $M_{i,0}$ 產生輔助金匙及該群組之群組密鑰。

以下即就分別針對成員加入，離開群組的密鑰管理機制做描述。

1. 成員加入密鑰管理機制

某一節點通過身份認證後，叢集頭必須有義務協助它成為叢集的成員之一。因此，就叢集頭而言，它必須執行成員加入機制。然而在 EBS

機制中，並未提及多人加入之密鑰管理機制，然而在無基礎行動網路下，此種情況會經常發生，即可能同一時間有多人進行加入群組的動作。若個別以單人加入法逐一加入成員，則需一直變動 Bit String 表格，且叢集頭每處理一個新加入的成員，就需產生新的群組密鑰加密後送出，造成叢集頭運算能力及網路頻寬的浪費。

因此本論文延伸原先 EBS 單人加入機制，提出能同時支援單人與多人同時加入之演算法，以下即詳述此成員加入演算法並詳加舉例。

成員加入演算法分為三步驟：

(1)叢集頭判斷是否增加輔助金匙；

(2)分配 Key String 和對應的輔助金匙給對應的新加入成員。

(3)系統需記錄「金匙變動點」;

(4)叢集頭送出新子叢集密鑰與輔助金匙訊息。

以下針對此三步驟做詳細描述：

(1)叢集頭判斷是否增加輔助金匙：

假設共有 x 位新成員欲加入此群組，加入的後的群組數目為 n+x，叢集頭需先判斷目前的輔助金匙數目是否滿足 $\binom{k+m}{k}$ 仍大於或等於 n+x，若此條件滿足，則僅需分配新的 Key String 和對應的輔助金匙給對應的新加入成員。反之，若 $\binom{k+m}{k}$ 小於 n+x，則需增加輔助金匙的數目，使新的 $\binom{k+m}{k}$ 大於或等於 n+x。

換句話說，此方法不需針對每位成員的加入修改 Bit String 表格，而是一次增加足夠的輔助金匙，並且分配 Bit String 給所有新加入的成員。

(2)分配 Key String 和對應的輔助金匙給對應的新加入成員。

若需增加輔助金匙，叢集頭可如同 EBS 自行判斷欲增加每個人

握有的輔助金匙數或是增加 rekey 訊息的送出次數以滿足$\binom{k+m}{k}$大於或等於 n+x，即：

A. 增加每個人握有的輔助金匙數：

即除了新加入成員以外的其他所有成員，都取得新增加的輔助金匙A_i。則子叢集頭將先前的 k+m 把輔助金匙（不含新產生的輔助金匙）任選 k+1 把分配給新加入成員。

B. 增加 rekey 訊息的送出次數：

即除了新加入成員取得新增加的輔助金匙A_i，其他所有成員握有的輔助金匙都維持不變。因此子叢集頭需將先前的 k+m 把輔助金匙（不含新產生的輔助金匙）任選 k-1 把分配給新加入成員。

本論文為簡化叢集頭的判斷流程，故設計若新增的輔助金匙編號為單數，則採用 A 步驟；反之若新增的輔助金匙編號為雙數，則採用 B 步驟。

(3)系統需記錄「金匙變動點」：

但系統增加輔助金匙後，不管是決定要增加每個人握有的輔助金匙數(增加 k)，或是增加 rekey 訊息的送出次數(增加 m)，都需留下記錄(需記錄哪位成員的加入，與其加入前的 Bit String 表格)，即找出哪些成員的加入是造成輔助金匙增加的臨界點，本論文稱這些成員為「金匙變動點」。

因此當成員離開時即需此記錄來幫助系統回復到最佳的狀況，使成員離開的運作上更有效率。

(4)叢集頭送出新子叢集密鑰與輔助金匙訊息：

當 Bit String 表格成功建立後，即可由叢集頭以舊子叢集密鑰加密新的子叢集密鑰，送出給先前的成員，若先前成員應握有的

子叢集輔助金匙有增加，則亦需以舊子叢集密鑰加密先前成員應擁有的子叢集輔助金匙並送給先前成員；並且以新成員和叢集頭共享的私密金鑰，加密新的輔助金匙和新子叢集密鑰後送出給新成員。

	M_{i0}	M_{i1}	M_{i2}	M_{i3}	M_{i4}	M_{i5}
A_{i1}	1	0	1	1	0	0
A_{i2}	0	1	1	0	1	0
A_{i3}	1	1	0	0	0	1
A_{i4}	0	0	0	1	1	1

<單人加入-輔助金匙數目不變>

成員 M_{i5} 加入叢集 i，CH_i 需送出的加密訊息：

$E_{Sub-GK_i}\{\text{Sub-GK}_i'\}$ -> $M_{i0} \sim M_{i4}$

$E_{SPW_5}\{\text{Sub-GK}_i', A_{i3}, A_{i4}\}$ -> M_{i5}

圖 3-2：成員 M_{i5} 加入叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

	M_{i0}	M_{i1}	M_{i2}	M_{i3}	M_{i4}	M_{i5}	M_{i6}
A_{i1}	1	0	1	1	0	0	1
A_{i2}	0	1	1	0	1	0	1
A_{i3}	1	1	0	0	0	1	1
A_{i4}	0	0	0	1	1	1	0
A_{i5}	1	1	1	1	1	1	0

<單人加入-輔助金匙數目增加>

成員 M_{i6} 加入叢集 i，CH_i 需送出的加密訊息：

$E_{Sub-GK_i}\{\text{Sub-GK}_i'\}$ -> $M_{i0} \sim M_{i5}$

$E_{SPW_6}\{\text{Sub-GK}_i', A_{i1}, A_{i2}, A_{i3}\}$ -> M_{i6}

圖 3-3：成員 M_{i6} 加入叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

舉例說明：以單人加入為例，若成員 M_{i5} 想加入叢集 i，則 Bit String 表格可如圖 3-2. 所示。即表示叢集頭 CH_i 需將新的子叢集密鑰 $\text{Sub-GK}_i'$ 以舊群組密鑰加密後，送給 $M_{i0} \sim M_{i4}$，並且以 SPW_5 加密新群組密鑰和 M_{i5} 應握有的輔助金匙（A_{i3}, A_{i4}）送給新成員 M_{i5}，如圖 3-2 所示。但因為 M_{i5} 加入時輔助金匙並未增加，因此不需留下記錄。

另若成員 M_{i6} 亦想加入叢集 i，需增加輔助金匙 A_{i5}，滿足在 EBS 所提及 $\binom{k+m}{k}$ 必須大於等於 n 的條件，且因為新增的輔助金匙 A_{i5} 編號為單

數，故採用前述的 A 步驟，即增加每人握有的輔助金匙數 k。Bit String 表格可如圖 3-3 所示。叢集頭 CH_i 同樣的需將新的群組密鑰 Sub-GK$_i$' 以舊群組密鑰加密後，送給 M_{i0} ~ M_{i5} ，並且以 SPW_6 加密新群組密鑰和 M_{i6} 應握有的輔助金匙(A_{i1}, A_{i2}, A_{i3})送給新成員 M_{i6} ，如圖 3-3 所示。因為 M_{i6} 加入時輔助金匙增加，因此需留下記錄，內容為成員 M_{i6} 加入前的 Bit String 表格，即圖 3-2 的 Bit String 表格。

	M_{i0}	M_{i1}	M_{i2}	M_{i3}	M_{i4}	M_{i5}	M_{i6}
A_{i1}	1	0	1	1	0	0	1
A_{i2}	0	1	1	0	1	0	1
A_{i3}	1	1	0	0	0	1	1
A_{i4}	0	0	0	1	1	1	0
A_{i5}	1	1	1	1	1	1	0

<多人加入-輔助金匙數目增加>

成員 M_{i3} 、M_{i4} 、M_{i5} 、M_{i6} 加入叢集 i ，CH_i 需送出的加密訊息：

E_{Sub-GK_i} { Sub-GK$_i$', A_{i4} } -> M_{i0} ~ M_{i2}

E_{SPW_3} { Sub-GK$_i$', A_{i1}, A_{i4}, A_{i5} } -> M_{i3}

E_{SPW_4} { Sub-GK$_i$', A_{i2}, A_{i4}, A_{i5} } -> M_{i4}

E_{SPW_5} { Sub-GK$_i$', A_{i3}, A_{i4}, A_{i5} } -> M_{i5}

E_{SPW_6} { Sub-GK$_i$', A_{i1}, A_{i2}, A_{i3} } -> M_{i6}

圖 3-4：成員 M_{i3} ~ M_{i6} 加入叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

另若成員多人加入，亦可採用此演算法進行。

舉例說明：若成員 M_{i3}、M_{i4}、M_{i5}、M_{i6} 同時想加入叢集 i，需增加輔助金匙 A_{i4}、A_{i5}，才可滿足在 EBS 所提及 $\binom{k}{m}$ 必須大於等於 n 的條件。 Bit String 表格可如圖 3-4. 所示。叢集頭 CH_i 同樣的需將新的子叢集密鑰 Sub-GK^i’ 以舊子叢集密鑰加密後，送給 M_{i0}~M_{i2}，且根據 Bit String 表格，M_{i0}~M_{i2} 需握有 A_{i5}，因此將此輔助金匙和新子叢集密鑰以舊子叢集密鑰加密後送出給 M_{i0}~M_{i2}。並且分別以每位新成員和叢集頭共享的私密金鑰加密新子叢集密鑰和個別新成員應握有的輔助金匙 A_{ik} 送給新成員 M_{i3}~M_{i6}，如圖 3-4 所示。

2. 成員離開密鑰管理機制

若有某一成員或某一群成員離開此叢集時，叢集頭除了必須依照 EBS 演算法驅離此叢集內需離開的成員外，亦須藉由 sum of product 化簡機制將新的子叢集密鑰以卡諾圖化簡後結果的子叢集輔助金匙之組合加密後送至未離開的節點手中。最後，再將不應該驅離的成員加回到此叢集。

而成員離開在 EBS 中亦有單人離開的機制，而無提及多人同時離開群組的情況，但在無基礎行動網路下，此種情況會經常發生，例如因為無線設備訊號的變化，便可能同一時間有多人離開群組的動作。若個別以單人離開法逐一做成員驅逐的步驟，則需一直變動 Bit String 表格，且叢集頭每處理一個離開的成員，就需產生新的子叢集密鑰加密後送出，造成叢集頭運算能力及網路頻寬的浪費，而每位成員亦需浪費其運算能力在輔助金匙的計算上。

因此本論文乃提出可支援多人同時離開的密鑰管理機制，此機制延伸 EBS 的單人離開演算法，並且也結合卡諾圖協助運算，快速找出可用來加密訊息的輔助金匙，避免被驅逐的所有成員解開此訊息，成員加入

演算法分為四步驟：

(1)將欲離開的成員(令其為 X 集合，離開的成員數為 x 人)，和後面加入的 x 位成員(令其為 Y 集合)共同驅逐出此群組，換句話說，即將(X∪Y)集合中的所有成員共同驅逐 (所有的A_i都會藉由雜湊函數被更新)；

(2)使用卡諾圖化簡找出可用來加密新子叢集密鑰的輔助金匙，以這些輔助金匙加密新的子叢集密鑰後送出，而可解開此訊息的成員則以$f(\text{Sub-GK}_i', A_i) = A_i'$更新其握有的$A_{ik}$，叢集頭亦以此雜湊函數更新所有的$A_{ik}$。

(3)若 Y 集合中，某些成員加入時系統有留下輔助金匙變化的紀錄，則系統還原至編號最小的成員加入前的狀態；

(4)依序增加(Y-X)集合中的成員回此群體，即是將不應被驅逐的成員加回此群體，但在(Y-X)集合中的成員會被分配先前離去成員 X 的 Key String 和新的對應A_i。

舉例說明：若成員M_{i2}想離開叢集i，假設原先的 Bit String 表格如圖 3-5.之原始 Bit String 表格所示。依本論文提出的成員離開演算法，X=$\{M_{i2}\}$、Y=$\{M_{i4}\}$，因此 X∪Y 集合中的成員為$\{M_{i2}, M_{i4}\}$，所以同時驅逐M_{i2}和M_{i4}，所有的輔助金匙A_i都會藉由雜湊函數$f(\text{Sub-GK}_i', A_{ik})$被更新。

但當同時驅逐M_{i2}和M_{i4}時，在 EBS 並未提及要採用哪些輔助金匙加密新的子叢集密鑰訊息，因此本論文結合卡諾圖的計算，快速找出可用來加密的輔助金匙，避免被驅逐的M_{i2}和M_{i4}解開此訊息，需注意的是，因為本論文採用的輔助金匙皆為正元素，因此卡諾圖計算出的結果中，若項數含有反元素，例如$\overline{A_i}$，在本論文中定義為無意義。因此結果如圖 3-5.之卡諾圖所示。

本論文採用雜湊函數協助更新輔助金匙，即取得新的子叢集密鑰成員可和其原先握有的輔助金匙當作輸入，經由雜湊函數的運算得到新的

輔助金匙，可表示為 $f(\text{Sub-GK}_i', A_i) = A_i'$，因為叢集頭不需傳送更新的輔助金匙，而交由各個成員自行運算，因此可減少網路傳送訊息的大小。

另外因 M_{i4} 加入時系統並未留下造成輔助金匙變化的紀錄，因此不需做還原的的動作，直接刪除 M_{i4} 此欄即可。接著需將不應被驅逐的成員 M_{i4} 加回此群組，即執行前述的加入演算法，但成員 M_{i4} 的 Bit String 會被分配離開成員 M_{i2} 的 Bit String 和對應的輔助金匙 A_i，換句話說，成員 M_{i4} 會被視為新的 M_2。需注意的是若原先 M_{i2} 加入時，在系統中有留下金匙變動點的紀錄，則 M_{i4} 會繼承此金匙變動點的身份，即當 M_{i4} 離開，需將系統還原至此筆記錄原先 M_2 加入前的系統狀態。最終的 Bit String 表格與叢集頭應送出的訊息如圖 3-5 之更新後 Bit String 表格所示。

另若成員 M_{i1} 亦想離開叢集 i，需減少輔助金匙 A_{i4}，因為三把輔助金匙即滿足在 EBS 所提及 $\binom{k+m}{k}$ 必須大於等於 n 的條件，避免輔助金匙的浪費。原先的 Bit String 表格如圖 3-6. 之原始 Bit String 表格所示。X=$\{M_{i1}\}$、Y=$\{M_{i3}\}$，因此 X∪Y 集合中的成員為$\{M_{i1}, M_{i3}\}$，所以將 M_{i1} 和 M_{i3} 同時驅逐，而所有的輔助金匙 A_i 都會藉由雜湊函數被更新。同樣的透過卡諾圖的計算，快速找出可用來加密的輔助金匙，避免被驅逐的 M_{i1} 和 M_{i3} 解開此訊息，如圖 3-6. 之卡諾圖所示。系統還原至 M_{i3} 加入前的狀態，即還原至 M_{i3} 加入前的 Bit String 表格，因為 M_{i3} 加入前，在 Bit String 表格中僅有三把輔助金匙，因此會減少一把輔助金匙，即是將 Bit String 表格中的最後一列刪掉。接著需將不應被驅逐的成員 M_{i3} 加回此群組，執行加入演算法，成員 M_{i3} 的 Bit String 分配給 M_{i1} 的 Bit String 和對應的輔助金匙 A_{i2} 和 A_{i3}。最終的 Bit String 表格與叢集頭應送出的訊息如圖 3-6. 之更新後 Bit String 表格所示。

舉例說明：以多人同時離開的運作為例，若成員M_{i1}和M_{i3}想離開叢集i，需減少輔助金匙A_{i4}，因為三把輔助金匙即滿足在 EBS 所提及$\binom{k+m}{k}$必須大於等於 n 的條件，避免輔助金匙的浪費。原先的 Bit String 表格如圖 3-7. 之原始 Bit String 表格所示。如上述的步驟所示，X={M_{i1}, M_{i3}}、Y={M_{i3}, M_{i4}}，因此將 X∪Y 集合中的成員{M_{i1}, M_{i3}, M_{i4}}同時驅逐出此群體，而所有的輔助金匙A_{ij}都會藉由雜湊函數被更新。。同樣的透過卡諾圖的計算，快速找出可用來加密的輔助金匙，避免被驅逐的M_{i1}和M_{i3}解開包含新子叢集密鑰的訊息。系統還原至M_{i3}加入前的狀態，即還原至M_{i3}加入前的 Bit String 表格，因為M_{i3}加入前，在 Bit String 表格中僅有三把輔助金匙，因此會減少一把輔助金匙，即是將 Bit String 表格中的最後一列刪掉，如圖 3-7 之更新後 Bit String 表格所示。

接著需將不應被驅逐的成員M_{i4}加回此群組，執行單人加入演算法，成員M_{i4}的 Bit String 分配給M_{i1}的 Bit String 和對應的輔助金匙A_{ij}。最終的 Bit String 表格與叢集頭應送出的訊息如圖 3-7 之更新後 Bit String 表格所示。

原始 Bit String 表格					
	M_{i0}	M_{i1}	M_{i2}	M_{i3}	M_{i4}
A_{i1}	1	0	1	1	0
A_{i2}	0	1	1	0	1
A_{i3}	1	1	0	0	0
A_{i4}	0	0	0	1	1

卡諾圖				
$A_{i1}\,A_{i2}\,/\,A_{i3}\,A_{i4}$	00	01	11	10
00	X	X	X	X
01	X	0	X	1
11	0	X	X	X
10	X	1	X	1
得到結果= $A_{i3}+A_{i1}.\,A_{i4}$				

更新後 Bit String 表格				
	M_{i0}	M_{i1}	M_{i2}（原先的 M_{i4}）	M_{i3}
A_{i1}	1	0	1	1
A_{i2}	0	1	1	0
A_{i3}	1	1	0	0
A_{i4}	0	0	0	1

<單人離開-輔助金匙數目不變>

成員 M_{i2} 離開叢集 i，CH_i 需送出的加密訊息：

1. 將 M_{i2} 和 M_{i4} 驅逐出此群體

$E_{A_{i3}}\{\text{Sub-GK}_i\text{'}\}$ -> M_{i0} 、M_{i1} M_{i3}

$E_{A_{i1}}\{E_{A_{i4}}\{\text{Sub-GK}_i\text{'}\}\}$ -> M_{i3}

注意：收到新群組密鑰者需以雜湊函數更新其握有的輔助金匙。

2. 系統回復至 M_{i4} 加入前的的狀態

3. 增加成員 M_{i4} 回此群體

$E_{Sub-GK_i}\text{'}\{\text{Sub-GK}_i\text{''}\}$ -> M_{i0} 、M_{i1} 、M_{i3}

$E_{SPW_4}\{\text{Sub-GK}_i\text{''}, A_{i1}\text{'}\ ,\ A_{i2}\text{'}\}$ -> M_{i4}

注意：A_{i1}' 和 A_{i2}' 乃在第一步驟中即由每位成員和叢集頭自行完成更新，因此此處叢集頭送出新的輔助金匙給 M_{i4}。

圖 3-5：成員 M_{i2} 離開叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

原始 Bit String 表格				
	M_{i0}	M_{i1}	M_{i2}	M_{i3}
A_{i1}	1	0	1	1
A_{i2}	0	1	1	0
A_{i3}	1	1	0	0
A_{i4}	0	0	0	1

卡諾圖				
$A_{i1}\ A_{i2}$ / $A_{i3}\ A_{i4}$	00	01	11	10
00	X	X	X	X
01	X	X	X	0
11	1	X	X	X
10	X	0	X	1
得到結果= $A_{i1}.A_{i2}+A_{i1}.A_{i3}$				

更新後 Bit String 表格			
	M_{i0}	M_{i1}（原先的 M_{i3}）	M_{i2}
A_{i1}	1	0	1
A_{i2}	0	1	1
A_{i3}	1	1	0

<單人離開-輔助金匙數目減少>

成員 M_{i1} 離開叢集 i，CH_i 需送出的加密訊息：

1. 將 M_{i1} 和 M_{i3} 驅逐出此群體

$E_{A_{i1}}\{E_{A_{i2}}\{\text{Sub-GK}_i'\}\}$ -> M_{i2}

$E_{A_{i1}}\{E_{A_{i3}}\{\text{Sub-GK}_i'\}\}$ -> M_{i0}

注意：收到新群組密鑰者需以雜湊函數更新其握有的輔助金匙。

2. 系統回復至 M_{i3} 加入前的的狀態

3. 增加成員 M_{i3} 回此群體

$E_{Sub-GK_i}'\{\text{Sub-GK}_i''\}$ -> M_{i0} 、 M_{i2}

$E_{SPW_3}\{\text{Sub-GK}_i''$, A_{i2}' , $A_{i3}'\}$ -> M_{i3}

注意：A_{i2}' , A_{i3}' 乃在第一步驟中即由每位成員和叢集頭自行完成更新，因此此處叢集頭送出新的輔助金匙給 M_{i3}。

圖 3-6：成員 M_{i1} 離開叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

原始 Bit String 表格					
	M_{i0}	M_{i1}	M_{i2}	M_{i3}	M_{i4}
A_{i1}	1	0	1	1	0
A_{i2}	0	1	1	0	1
A_{i3}	1	1	0	0	0
A_{i4}	0	0	0	1	1

卡諾圖				
$A_{i1}\ A_{i2}$ / $A_{i3}\ A_{i4}$	00	01	11	10
00	X	X	X	X
01	X	0	X	0
11	1	X	X	X
10	X	0	X	1
得到結果= $A_{i1}.\ A_{i3} + A_{i1}.\ A_{i2}$				

更新後 Bit String 表格			
	M_{i0}	M_{i1}（原先的 M_{i4}）	M_{i2}
A_{i1}	1	0	1
A_{i2}	0	1	1
A_{i3}	1	1	0

<多人離開-輔助金匙數目減少>

成員 M_{i1}、M_{i3} 離開叢集 i，CH_i 需送出的加密訊息：

1. X={ M_{i1}, M_{i3} }，Y={ M_{i3}, M_{i4} }，因此將 X∪Y 集合中的成員{ M_{i1}, M_{i3}, M_{i4} }同時驅逐出此群體

$E_{A_{i1}}\{E_{A_{i3}}\{\text{Sub-GK}_i'\}\} \rightarrow M_{i0}$

$E_{A_{i1}}\{E_{A_{i2}}\{\text{Sub-GK}_i'\}\} \rightarrow M_{i2}$

注意：收到新群組密鑰者需以雜湊函數更新其握有的輔助金匙。

2. 系統回復至 M_{i3} 加入前的的狀態

3. 增加成員 M_{i4} 回此群體

$E_{Sub-GK_i}'\{\text{Sub-GK}_i''\} \rightarrow M_{i0}$、$M_{i2}$

$E_{SPW_4}\{\text{Sub-GK}_i'', A_{i2}', A_{i3}'\} \rightarrow M_{i4}$

注意：A_{i2}', A_{i3}' 乃在第一步驟中即由每位成員和叢集頭自行完成更新，因此此處叢集頭送出新的輔助金匙給 M_{i4}。

圖 3-7：成員 M_{i1} M_{i3} 同時離開叢集 i 之 Bit String 表格與 CH_i 送出的加密訊息

但須注意的是，若離開成員為叢集頭，需做重新初始的的動作，情況可分成以下兩種：

I. 子叢集中的叢集頭(M_{i0})離開此群組

需自叢集內成員再找出一位與其他成員都為 one hop 的成員擔任叢集頭，若沒有成員能滿足此條件，則此叢集會被迫瓦解，在此

叢集內的成員可嘗試加入其他叢集。新的叢集頭M_{i0}’會收到其他成員送出的私密金鑰，並產生新的子叢集輔助金匙和子叢集密鑰，以握有的每位成員的私密金鑰加密後送出給叢集內的成員。對於整個群組來說，則群組控制者只需進行離開/加入演算法各一次，驅逐M_{i0}並加入新的M_{i0}’，即可維持整個安全群播環境正常運作。

II. 群組控制者(M_{00})離開群組

需自所有子叢集頭中再找出一位擔任新的群組控制者(我們假設編號最小者擔任新的群組控制者)。新的群組控制者會收到其他子叢集頭送出的私密金鑰後，產生新的輔助金匙和群組密鑰，以握有的每位子叢集頭的私密金鑰加密後送出給群組內的子叢集頭。

以上即是本論文延伸 EBS 並結合卡諾圖，所得的叢集架構下可採用之群播密鑰管理機制。

3.2.3 群播資料傳送

根據上述本論文提出的密鑰管理機制，本論文探討叢集頭如何協助該叢集之成員叢集內、叢集間與跨叢集群播之目的。以下即就每一種情況做描述。

1. 叢集內群播資料傳送

當某一叢集內成員欲進行群播時，其步驟如下：

i. M_{ij}必須決定群播對象有哪些。

ii. M_{ij}將此群播的訊息藉由先前和叢集頭共享的私有密鑰SPW_j，使用對稱式密碼系統加密，並傳送至此叢集頭M_{i0}。此M_{ij}所需加密與傳送的內容，包括訊息 M，及群播對象的 ID。ID 可為叢集內某些成員的成員編號或是”ALL”(代表叢集內所有成員)。

iii. 叢集頭M_{i0}解開訊息後，便依據M_{ij}所要求的群播對象進行群播。若 ID 為”ALL”，則以叢集內的子叢集密鑰(即

Sub-GK$_i$)，以對稱式加密後送出；若是子叢集內的部分成員，此時M_{i0}會先藉由 sum of product 算出最後須由哪幾把子叢集輔助金匙須相互合作，以達訊息群播之目的。

iv. 任一節點收到此群播訊息後，由於它是屬於合法的接收成員，故可用它手中的子叢集密鑰或是子叢集輔助金匙進行組合以解開此訊息。

2. 叢集間群播資料傳送

而當某一叢集頭M_{i0}欲進行群播時，其作法與叢集內做法相似，步驟如下：

i. M_{i0}必須決定群播對象有哪些。

ii. M_{i0} 將此群播的訊息藉由先前和叢集頭共享的私有密鑰PW_i，使用對稱式密碼系統加密，並傳送至此叢集頭M_{00}。此M_{i0}所需加密與傳送的內容，包括訊息 M，及群播對象的 ID。ID 可為叢集間某些叢集頭的成員編號或是”ALL”（代表叢集間所有叢集頭）。

iii. 叢集頭M_{00}解開訊息後，便依據$M_{i,0}$所要求的群播對象進行群播。若 ID 為”ALL”，則以叢集間的群組密鑰(即 GK)，以對稱式加密後送出；若僅包含部分的叢集頭，此時M_{00}會先藉由 sum of product 算出最後須由哪幾把群組輔助金匙須相互合作，以達訊息群播之目的。

iv. 任一叢集頭收到此群播訊息後，由於它是屬於合法的接收成員，故可用它手中的群組密鑰或是輔助金匙進行組合以解開此訊息。

3. 叢集內與叢集間混合之群播資料傳送

假設某一叢集CH_i之某一節點M_{ij}欲執行跨叢集的群播，此時的群播機制做法如下：

i. M_{ij}必須決定群播對象有哪些。

ii. M_{ij} 將此群播的訊息藉由先前和叢集頭共享的私有密鑰

SPW_j，使用對稱式密碼系統加密，並傳送至此叢集頭M_{i0}。此節點M_{ij}所需加密與傳送的內容，包括訊息 M，群播對象的所有節點之 ID 及其相對應叢集 ID。

iii. 叢集頭M_{i0}解開訊息後，便依據M_{ij}所要求的群播對象轉成自己的需求，並重複叢集間的群播機制。

iv. 叢集頭M_{00}解開訊息後，它會依照訊息內容決定採用群組密鑰或是藉由 sum of product 化簡機制找出符合的輔助金匙，將此群播訊息轉至那些節點所屬的叢集之叢集頭手中。

v. 任一叢集頭收到此群播訊息後，由於它是屬於合法的接收成員，故可用它手中的群組密鑰或是輔助金匙進行組合以解開此訊息，然後再利用叢集內的群播機制送到必須收到此群播訊息的成員手中。以上即是叢集內與叢集間混合之安全群播機制。

3.2.4 與 EBS 的方法比較

此小節主要是針對本論文在 3.1 節中所分析的問題做檢驗，檢驗本論文所提的方法是否改進了該問題，以下驗證本論文所提出的密鑰管理協定是否可以解決 3.1 節所分析出的問題：

- 結合卡諾圖協助群組訊息傳送

  在本論文所提的方法中，加入卡諾圖協助系統找出正確的加密輔助金匙，僅讓應收到訊息的成員能解開此訊息。但卡諾圖應用上，會有反元素的產生，輔助金匙的反元素在本論文中並未有定義，本論文採用的輔助金匙皆為正元素，因此卡諾圖計算出的結果中，若項數含有反元素，例如$\overline{A_i}$，在本論文中定義為無意義，此乃本論文為結合卡諾圖與 EBS 機制所做的其中一部份調整。

  舉例說明：如圖 3-8 所示，可以看出卡諾圖對輔助金匙的運算，並

非只有一固定解，但在這些答案中，我們必須挑選一個沒有反元素的輔助金匙解，用來加密訊息。

- 延伸 EBS 機制設計多人加入/離開演算法

  本論文延伸 EBS 機制未提及的多人加入/離開演算法，使成員能有效率的進行加入與離開的動作，多人加入與離開的的演算法如圖 3-9 所示。

- 叢集架構下的安全群播

  本論文在叢集架構下，結合本論文提及的密鑰管理機制，進行群組訊息的安全群播，達成安全群播的其中三要件，如 4.1.1 節所示。

卡諾圖				
$A_{i1}\,A_{i2}\,/\,A_{i3}\,A_{i4}$	00	01	11	10
00	X	X	1	X
01	X	0	X	0
11	X	X	X	X
10	X	1	X	0
得到結果(1)= $A_{i1}.\,A_{i4}+A_{i3}.\,A_{i4}$				
得到結果(2)= $\overline{A_{i1}}.\,\overline{A_{i2}}+A_{i3}.\,A_{i4}$				
得到結果(3)= $\overline{A_{i1}}.\,\overline{A_{i2}}+A_{i1}.\,\overline{A_{i3}}$				
得到結果(4)= $A_{i1}.\,A_{i4}+A_{i1}.\,\overline{A_{i3}}$				

圖 3-8：卡諾圖運算

可支援多人加入演算法

1. 叢集頭判斷是否增加輔助金匙，若目前輔助金匙數目不夠，則直接增加足夠的量，滿足$\binom{k+m}{k}$大於或等於新群組成員數；
2. 分配Key String和對應的輔助金匙給對應的新加入成員。
3. 系統記錄「金匙變動點」，即找出哪些成員的加入是造成輔助金匙增加的臨界點；
4. 叢集頭送出新群組密鑰與輔助金匙。

可支援多人離開演算法

1. 令欲離開的成員為X集合，離開的成員數為x人，後面加入的x位成員為Y集合，將(X∪Y)集合中的所有成員共同驅逐；
2. 使用卡諾圖化簡找出可用來加密新群組密鑰的輔助金匙，以這些輔助金匙加密新群組密鑰密鑰後送出，而可解開此訊息的成員則以$f(\text{Sub-GK}_i', A_i) = A_i'$更新其握有的$A_{ik}$，叢集頭亦以此雜湊函數更新所有的$A_{ik}$。
3. 若Y集合中第一位成員加入時系統有留下輔助金匙變化的紀錄，則系統還原至此成員加入前的狀態；
4. 依序增加(Y-X)集合中的成員回此群體，但分配先前離去成員X的Key String和新的對應A_i。

圖3-9：多人加入/離開演算法

四・比較與分析

在第三章中，本論文基於 EBS 的方法，提出了一個可支援多人加入/離開的密鑰管理協定，主要是針對本論文所分析出的問題做改進；此章則進一步對本論文所提出的方法做分析與比較，主要包括安全及效率兩方面。

4.1 安全性的比較與分析

由於本論文所探討的密鑰管理協定是運用在動態群組的環境上，即成員可隨時動態地加入或離開群組，而綜合所看過的文獻資料[17]指出，基於安全性的考量，一個動態會議金鑰協定必須滿足幾個重要條件，因此，在本小節中，先對本論文所提出的方法做驗證，證明此協定符合該安全條件。

以下幾點是一個動態群組達成安全群播所應具有的安全條件，且各個條件之間並非相互獨立，而是互有關聯的，例如前推與後推安全性同時也必須包含群組訊息的安全[17]，因此本論文驗證所提出的密鑰管理架構是否滿足達成安全群播的此四項要件：

(1) 關於私密性(Confidentiality)安全方面，本論文所提的方法是以輔助金匙對欲傳送的訊息做加密，且唯有擁有對應輔助金匙的成員才能解開此訊息，對於一個非法的使用者而言，雖然可以取得群組成員所送之封包，但由於其不知道輔助金匙，因此仍無法解出其中所含的群播訊息。

(2) 關於身份認證(Authentication)方面，因身份認證並非本論文欲探討的部分，故在本論文中皆假設加入的成員為合法使用者。

(3) 關於前推安全性(Forward Security)與後推安全性(Backward Security)方面，必需同時檢視成員加入及離開時所採用的演算法，由前幾章的介紹可以了解到本論文所提的方法是建立在 EBS 方法中的架構之上，且結合

叢集架構與卡諾圖，設計完整的成員加入及離開演算法，來處理動態群組環境中成員加入及離開的動作，同步更新群組密鑰，以達到前推安全性與後推安全性。

4.2 效率性的比較與分析

由於本論文所提的協定之運作環境為無基礎行動網路，在此環境之下除了安全議題外，另外因無線終端設備的儲存空間與電力皆有限制，因此如何讓無線終端設備的使用更節省儲存空間與省電也是近起年來大家所努力的目標。換句話說，效率面也是一直是被大眾所重視的，包括需耗費的儲存空間、更新密鑰需送出的訊息次數、及運算效率，在合於安全性的要求之內，最理想的是整個協定每位成員需儲存的資訊和送出的 Rekey 訊息次數越少越好，而運算效率則愈快愈好。

因為本論文提出的密鑰管理架構係延伸 EBS 改進後所得，因此承襲 EBS 原先就有的優點：採用的輔助金匙個數和 rekey 訊息送出次數，明顯優於以二元樹資料結構來管理輔助金匙，將所需使用的輔助金匙和 rekey 訊息送出次數都維持最小值，且本論文所提出的多人加入/離開演算法更使 rekey 訊息較原先的 EBS 更少。而運算效率面我們結合卡諾圖，協助系統能快速找出加密的輔助金匙，使 Rekey 或是傳送群組資料的效率加快。根據以上所言，我們用以下幾個指標，衡量本論文提出的密鑰協定效率：

[儲存空間衡量]

- 輔助金匙數目(Number of Administrative Key)

  輔助金匙數目包含叢集頭應管理的所有輔助金匙數目，與每位成員應握有的輔助金匙數目。當在同樣的安全條件下，欲採用的輔助金匙數目越少，即表示每位成員需花費的儲存空間越小。

[傳輸效率衡量]

- Rekey 訊息量 (Rekey Message)

  當群組中有成員加入或離開時，系統皆須作群組密鑰更新的動作，送出的 Rekey 訊息數量少，可以節省無線網路上頻寬的浪費，及每位成員耗費在處理 Rekey 訊息的時間。

[運算效率衡量]

- 運算處理 (Computation)

  當進行更新群組密鑰時，叢集頭花費在處理 Rekey 訊息的時間越短，則可以增進整個流程的效率。而叢集頭花費在處理 Rekey 訊息的時間包含兩部分：產生 Rekey 訊息與找出用來加密的輔助金匙。而這個指標可以看出一個叢集頭在運算上的負擔，叢集頭能越有效的處理這些運算，對於整個系統的效率愈有幫助。

通常在一個群組密鑰協定中，密鑰更新階段是最費時費力的，因為群組的成員可能會經常加入或離開群體，系統即需頻繁的進行密鑰更新的動作。前述的效率衡量指標：Rekey 訊息量 (Rekey Message)對於系統的執行效率會有很大影響；且叢集頭在更新密鑰時，要找出適當的輔助金匙加密群組密鑰，若叢集頭需要管理大量的輔助金匙，亦會降低其進行密鑰更新時的速率。

一般衡量密鑰管理系統的效率皆著重在成員加入與離開的部分，因此本論文對於密鑰協定中密鑰更新的演算法，利用表 4-1 列出本論文所探討過的密鑰協定與本論文所提出的密鑰協定作成員加入/離開之效率比較：

表 4-1：群組密鑰協定之效率評估比較

演算法 / 衡量指標	傳統二元樹金匙管理	二階式群播金匙管理	EBS (僅有單人加入/離開演算法)	支援多人加入離開的密鑰管理機制
總輔助金匙數目	n-2	$2\lceil \log_2 n \rceil+1$	滿足 $\binom{k+m}{k} \geq$ n	滿足 $\binom{k+m}{k} \geq$ n
每人握有的輔助金匙數	$\lceil \log_2 n \rceil$-1	$\lceil \log_2 n \rceil$+1	k	k
x 位成員加入總 Rekey 訊息量	2 * x	2 * x	2 * x	x + 1
y 位成員離開總 Rekey 訊息量	y * $\lceil \log_2 n \rceil$	視卡諾圖化簡結果項數而定	m * y	m
採用卡諾圖		✓		✓

經由上述表 4-1 的效率評估比較，做了以下的解釋：

1. 在輔助金匙數目方面：

根據第二章所介紹的傳統二元樹及二階式群播金匙管理方法，我們知道傳統二元樹需要的總輔助金匙數目為 n-2，即不包含葉節點 (私有金鑰)及根節點(群組密鑰)的其他節點數目，如圖 2-2 所示。而二階式群播金匙管理方法則需 $2\lceil \log_2 n \rceil+1$，樹的每一階層有兩把輔助金匙 K_i 和 $\overline{K_i}$，詳見圖 2-5。

本論文所提方法的輔助金匙數目，較傳統二元樹或是二階式群播金匙管理來得少，所需的輔助金匙數目僅需滿足 $\binom{k+m}{k} \geq$ n 即可，表示輔助金匙數目僅需 $(k+m)$把。

以圖 4-1 來看，橫軸為群組成員個數，縱軸為此群組共需要的輔助金匙數量，隨著群組成員數目的增加，輔助金匙的數量也會增加，總輔助金匙數目以表 4-1 的公式計算得出，而 EBS 及本論文提出的密鑰管理機制，輔助金匙增加的幅度最小。所需的輔助金匙越少，表示叢集頭需耗費用來儲存輔助金匙的空間也會越少。

2. 在每人握有的輔助金匙數方面：

傳統二元樹每人需握有的輔助金匙數目為$\lceil \log_2 n \rceil$-1，即某成員 key path 上的輔助金匙個數，不包含葉節點 (私有金鑰)及根節點(群組密鑰)，如圖 2-2 所示。而二階式群播金匙管理方法每位成員則需握有$\lceil \log_2 n \rceil$+1 把輔助金匙，即某成員 key path 上的所有節點，詳見圖 2-5。

本論文提出的機制亦保有原先 EBS 機制下，每人僅需握有 k 把輔助金匙的優點，較傳統二元樹或是二階式群播金匙管理的$\lceil \log_2 n \rceil$來的少。舉例來看，當成員數目是 1024 人，若採用傳統二元樹或是二階式群播金匙管理，則共需 20 把以上的輔助金匙，而每人需握有 9 把以上的輔助金匙；但在 EBS 系統下，則共僅需 13 把輔助金匙，因$\binom{13}{6} = 1716 \geq 1024$，且每人僅需握有 6 把輔助金匙。因此本研究採 EBS 為本研究之根基。茲以

以圖 4-2 來看，橫軸為群組成員個數，縱軸為每位成員需要握有的輔助金匙數量，隨著群組成員數目的增加，每人需握有的輔助金匙數量也會增加，而 EBS 及本論文提出的密鑰管理機制，每人需握有的輔助金匙增加幅度最小。表示每位成員僅需耗費較少的儲存空間來儲存這些輔助金匙。

3. 在多人加入的群組 Rekey 訊息量方面：

假設有 x 位的成員同時加入，不管採用傳統二元樹、二階式群播金匙管理方法或是原先的 EBS 機制，都是僅以單人加入重複運作，則需送出的 Rekey 訊息為(2 * x)。因為每當一位成員加入，叢集頭就需送出新的群組密鑰給舊成員，並且另外以新成員和叢集頭共享的私有密鑰，加密新群組

密鑰和新成員應握有的輔助金匙，送給新成員，因此共需要的 Rekey 訊息為(2 * x)個。

但本論文提出的密鑰機制，因包含多人加入的演算法，因此不需要重複運作，僅需作一次成員加入的動作。因此需要的 Rekey 訊息為(x+1)個，包含(1)以舊群組密鑰加密新的群組密鑰和舊成員應握有的新增輔助金匙，送給舊成員以及(2)個別以加入新成員和叢集頭共享的私有密鑰，加密新群組密鑰和新成員應握有的輔助金匙，送給新成員，因此共需要的 Rekey 訊息僅為(x+1)個。

以圖 4-3 來看，我們假設目前群組成員個數為 50 人，橫軸為同時進行加入此群組動作的成員個數(即 x)，縱軸為叢集頭為更新群組密鑰所需送出的 Rekey 訊息數量，在這張圖中，我們發現傳統二元樹金匙管理、二階式群播金匙管理與 EBS 都需送出 2x 的訊息個數，但本論文提出的方法則僅需 x+1，Rekey 訊息數量遠低於其他三種方法，可以節省無線網路上頻寬的浪費，及每位成員耗費在處理 Rekey 訊息的時間。

4. 在多人離開的群組 Rekey 訊息量方面：

假設有 y 位的成員同時離開，若採用傳統二元樹，以單一成員離開(rekey 訊息數量為$\lceil \log_2 n \rceil$，如第二章所述)重複運作處理多人同時離開之情況，需 y * $\lceil \log_2 n \rceil$的 rekey 訊息數量。而二階式群播金匙管理方法則需視卡諾圖化簡結果項數決定需傳送的 rekey 訊息數量。

根據原先 EBS 的成員離開演算法所述，每當一位成員離開，叢集頭就需送出新的群組密鑰給未離開的成員，且要避免離開成員解開此訊息，因此需以離開成員缺少的輔助金匙加密新群組密鑰後送出，使留下來的成員皆能拿到新群組密鑰。且每次需將不應被驅逐的成員加回群組，所以 Rekey 訊息需加 2（包含送給舊成員和新加入成員的訊息，如第 3 點所述），但也有可能離開的成員剛好是最後加入的成員，就不需要做加入的動作，因此在效率分析此部分，省略此加入部分的 Rekey 訊息。故稱單人

離開的 Rekey 訊息為 m。

假設有 y 位的成員同時離開，若採用原先的 EBS 機制，即僅以單人離開重複運作，則需送出的 Rekey 訊息為(m * y)。但本論文提出的密鑰機制，因包含多人離開的演算法，因此不需要重複運作，僅需作一次成員離開的動作。因此需要的 Rekey 訊息為 m 個，即需送出 m 次的 rekey 訊息給其他成員，更新群組密鑰。故於輔助金匙數目上與 Rekey 訊息量上較其他架構來的有效率。

以圖 4-4 來看，我們假設目前群組成員個數為 50 人，橫軸為同時進行離開此群組動作的成員個數(即 y)，縱軸為叢集頭為更新群組密鑰所需送出的 Rekey 訊息數量。在表 4-1 中，我們知道傳統二元樹金匙管理的 Rekey 訊息數目為 y * $\lceil \log_2 n \rceil$，因我們假設 n=50，所以 Rekey 訊息數目為 y * $\lceil \log_2 50 \rceil$，表示 Rekey 訊息的數目會隨著同時離開成員的數目增加而變多；而 EBS 雖 Rekey 訊息數目較傳統二元樹金匙管理少，但仍須 m * y 個 Rekey 訊息數目，亦是隨著同時離開成員的數目增加而變多。但本論文提出的方法則僅需 m，表示 Rekey 訊息的數目不會隨著同時離開成員的數目增加而變多，大大節省無線網路上頻寬的浪費，及每位成員耗費在處理 Rekey 訊息的時間。

5. 在運算效率衡量上：

因本研究將 sum of product 的觀念與 EBS 結合，使得它在群播上及應用於單一節點或多個節點同時離開時，以卡諾圖快速找出可用來加密訊息的輔助金匙，因此能很快的將子叢集密鑰分配到未離開此叢集之群組成員手中，使整個系統的執行效率提升。

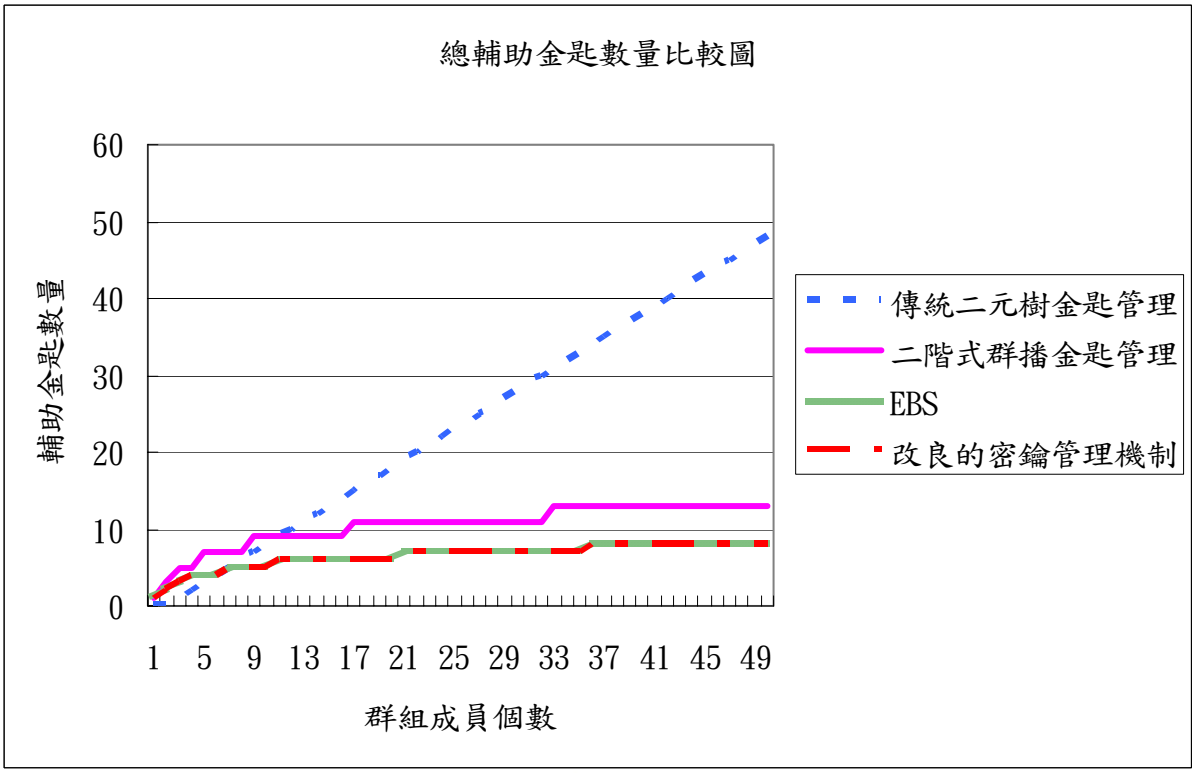


圖 4-1：總輔助金匙數量比較圖

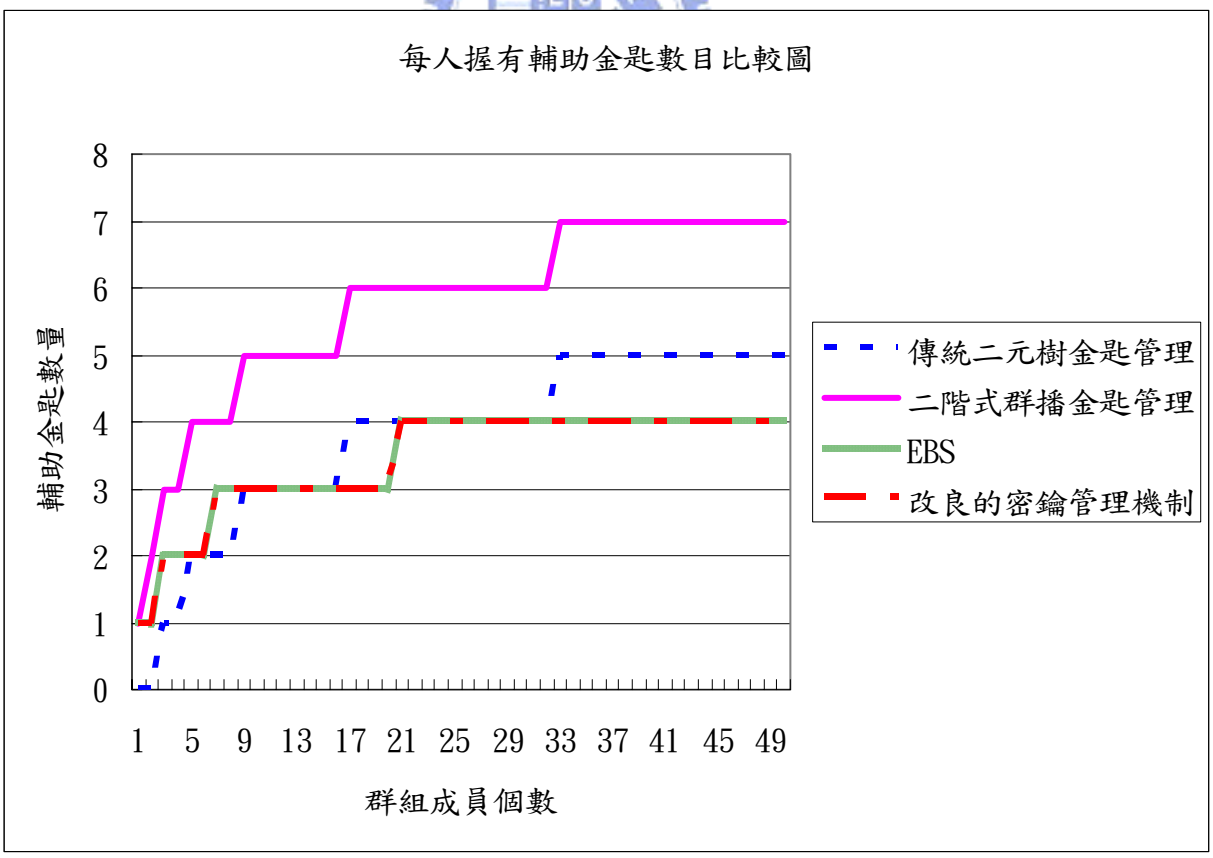


圖 4-2：每人握有輔助金匙比較圖

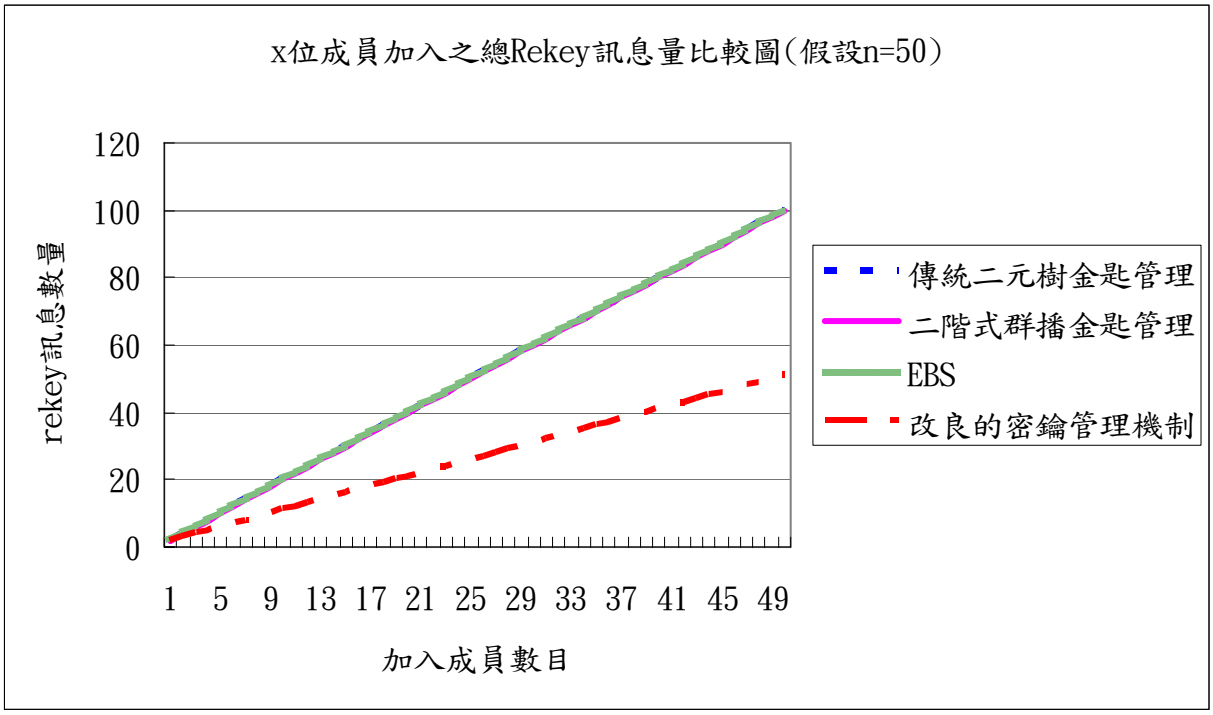


圖 4-3：假設 n=50, x 位成員加入之總 rekey 訊息數量比較圖

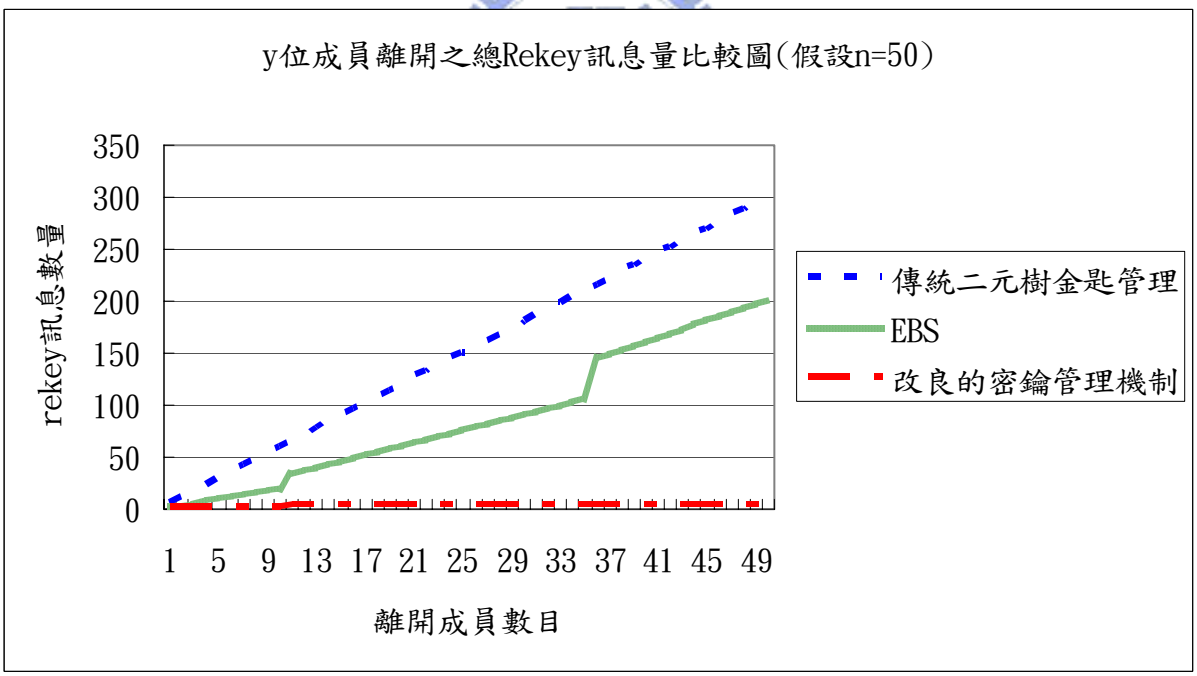


圖 4-4：假設 n=50, y 位成員離開之總 rekey 訊息數量比較圖

五・結論與未來研究方向

在本章中，對於本論文研究做個簡單的結論，說明本論文的貢獻，並且對未來的研究方向提出一些建議。

5.1 結論

經由第一章的研究動機說明，可以清楚了解本論文要做的方向，於是收集、研究了許多相關的文獻，如第二章所介紹，在探討的過程中，了解到 Ad Hoc 網路的特性、了解為什在本論文為何採用半集中的密鑰管理協定，而不採用分散式密鑰管理協定的架構，也了解利用 EBS 與卡諾圖進行運算的好處；在對過去學者所提的方法做完問題的分析之後，發現其仍存在某些安全及效率上的問題，於是本論文基於 EBS 方法架構之上，提出一個可支援多人同時加入/離開的密鑰管理協定，除了增加多人加入/離開的演算法之外，還增加卡諾圖運算機制以加強系統執行的效率，並在此密鑰管理機制下探討如何進行群播資料傳送。因無基礎行動網路因具有動態拓樸與自我組織之特性，使得它有別於其他無線網路架構，故使得它可應用的範圍更廣。而通訊過程的私密性、資料完整性、身份驗證及群播機制下的密鑰管理均是無基礎行動網路所需具備的安全機制。本研究在基於叢集架構下結合半集中式的密鑰管理機制，使加入的成員能安全的獲得輔助金匙，進而完成後續的金鑰管理協定，藉由 EBS 結合 sum of product，找出群播金鑰，最後透過架構好的群組密鑰管理架構完成安全群播。在安全性分析上，本論文滿足安全群播的三項安全條件，即可說明他具有足夠的安全性；另外在效率分析上，增加 EBS 方法缺乏的多人加入/離開機制，大大增加了系統執行效率上的強度，且利用卡諾圖運算找出加密的輔助金匙，增加運算上的效率。

5.2 未來研究方向

本論文假設想加入群組內的成員皆合法，忽略了成員身份認證的機制，但若是想要增加密鑰協定與安全群播環境的強健性(robust)，思考如何在此機制下建

立完善的成員身份認證方法也是一個值得研究的議題。且由於無基礎網路除具有無線網路的特性外，又加上本身的特質，使得它在安全機制的設計上需做改良，建議未來其他相關的安全議題，例如阻斷攻擊、存取控制等進行研究，如此，可提升該網路架構於應用上的安全性。

參考文獻

[1] J. Moy, “Link-state routing, in: Routing in Communications Networks,” ed. M.E. Steenstrup (Prentice Hall, 1995).

[2] G.S. Malkin, M.E. Steenstrup, “Distance-vector routing, in: Routing in Communications Networks,” ed. M.E. Steenstrup (Prentice Hall, 1995).

[3] C.E. Perkins, P. Bhagwat, “Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers,” in: Proc. Of ACM SIGCOMM’94, London, UK (August–September 1994) pp. 234–244.

[4] V.D. Park, M.S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in: Proc. of IEEE INFOCOM’97, Kobe, Japan (April 1997) pp. 1405–1413.

[5] M.S. Corson, A. Ephremides, “A distributed routing algorithm for mobile wireless networks,” Wireless Networks 1 (1995) 61–81.

[6] C.E. Perkins, E.M. Royer and S.R. Das, “Ad hoc on-demand distance vector (AODV) routing,” IETF MANET Working Group, Internet-Draft (March 2000).

[7] J. Broch, D.B. Johnson and D.A. Maltz, “The dynamic source routing protocol for mobile ad hoc networks,” IETF MANET Working Group, Internet-Draft (October 1999).

[8] M. Jiang, J. Li and Y.C. Tay, “Cluster based routing protocol (CBRP),” IETF MANET Working Group, Internet-Draft (August 1999).

[9] L. Zhou and Z. Haas, “Securing Ad Hoc Networks,” IEEE Network , pp.24-30, Dec, 1999.

[10] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, “Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks,” IEEE 9th International Conference on Network Protocols (ICNP'01), 2001.

[11] H. Luo and S. Lu, “Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks,” Technical Report TR-200030, Dept. of Computer Science, UCLA, 2000.

[12] L. Venkatraman and D. Agrawal, “A novel authentication scheme for ad hoc networks,” in IEEE Wireless Communications and Networking Conference (WCNC 2000), vol. 3, pp. 1268--1273, 2000.

[13] V. Varadharajan, R. Shankaran and M. Hitchens. “Security for cluster based ad

hoc networks," Computer Communications, 27(2004): 488-501.

[14] D. H. Tim, "The Cluster-Based Routing Protocol," project group 'Mobile Ad-Hoc Networks Based on Wireless LAN' winter semester 2003/2004.

[15] S. Rafeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," In ACM Computing Surveys, Vol.35, No.3, pp.309-329, September 2003.

[16] M. J. Moyer, J. R. Rao and P. Rohatgi, "A Survey of Security Issues in Multicast Communications," IEEE Network 13(6), Nov/Dec 1999, p.12-p.23.

[17] 陳惠淳，伍麗樵，"二階式群播金匙管理"， TANET'2000, 2000, p.24-p.31.

[18] L. Morales, I.H. Sudborough, M. Eltoweissy and M. H. Heydari, "Combinatorial Optimization of Multicast Key Management," proceedings of the 36th Hawali International Conference on System Sciences (HICSS'03), 2002.

[19] M. Conti and S. Giordano, "Mobile Ad-hoc Networking," In Proceedings of the 34th Hawail International Conference on System Sciences, 2001.

[20] D.M. Wallner, E.J. Harder, R.C. Agee, "Key Management for Multicast: Issues and Architectures", Informational RFC, draft-wallner-key-arch-00.txt, July 1997.