國 立 交 通 大 學

資訊管理研究所

碩 士 論 文

一個用於 Ad Hoc 無線網路上的改良式金鑰協同協定

An Improved Key Agreement Protocol for Ad Hoc Mobile Networks

研 究 生：黃 永 鑫

指導教授：羅 濟 群 教授

中 華 民 國 九十三 年 六 月

一個用於 Ad Hoc 無線網路上的改良式金鑰協同協定

An Improved Key Agreement Protocol for Ad Hoc Mobile Networks

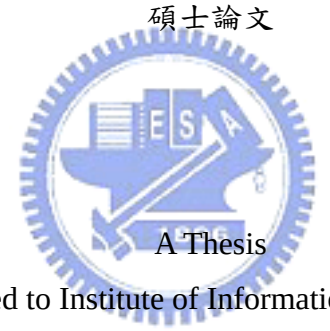
研 究 生：黃 永 鑫　　Student: Yong-Xin Huang

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 2004

Hsinchu, Taiwan, the Republic of China

中華民國　九十三 年 六 月

一個用於 Ad Hoc 無線網路上的改良式金鑰協同協定

研究生：黃永鑫　　指導教授：羅濟群 教授

國立交通大學資訊管理研究所

摘要

由於科技的發達，加上許多無線終端設備的普及、盛行，人們可以便利地利用網路來幫助資訊的流通，若想在 Ad Hoc 的無線網路環境上建立安全群組通訊環境，必需借助許多技術，其中密碼學的加解密方法，可以將資料加密後再送給合法的成員，使得非法的成員無法得知加密前的資訊，利用對稱式加密的方法配合資料群播的方式，便可更有效率地將資料傳給其它成員，但如何讓位於 Ad Hoc 環境上的群組成員能安全地得到同一支鑰呢？這便是本論文要探討的主題。

關於群體金鑰建立技術的金鑰協同協定(key agreement protocol)，雖然過去早已有許多學者提出其方法，但由於 Ad Hoc 網路環境的特性，必非所有的方法皆能完全適用，在本論文中做完問題分析之後，發現其仍存在某些安全及效率上的問題，因此本論文基於 Huang & Chang 方法架構上，提出一個改良後的金鑰協同協定，並且做了安全面及效率面的分析與比較，在安全方面，證明本論文所提的方法除了符合會議金鑰協定的安全條件之外，也解決了可能面臨的重送攻擊與密碼猜測攻擊，大大增加了安全上的強度；在效率方面，除了利用 XOR 運算取代模指數運算來產生會議金鑰以增加運算上的效率之外，本論文所提的方法中，單一傳輸訊息量並不會受成員數目的影響，而是單一固定量，除了可增加合作傳輸上的效率之外，還可減少傳輸訊息遺失、錯誤的問題。

關鍵字：Ad Hoc、金鑰協同協定

An Improved Key Agreement Protocol for Ad Hoc Mobile Networks

Student: Yong-Xin Huang Advisor: Dr. Chi-Chun Lo

Institute of Information Management

Nation Chiao Tung University

Abstract

With fast growth of Information Technology and the popularization of wireless device, people can communicate conveniently in the networks. Cryptology is an important technique for secure group communication. Using symmetrical encryption and multicast technique can help message transfer more efficiently. How to let group members in Ad Hoc network environment know the group key secretly? It's our paper object.

About key agreement protocol, many methods have been introduced. But not all of them are suitable for Ad Hoc network environment. The problems are maybe about security and efficiency. This paper introduces an improved method based on Huang & Chang's method. And we make analysis about the security and efficiency problem. About the security, the proposed method can conform to the conditions of secure key agreement protocol. Besides, it can avoid reply attack and password guessing attack. About efficiency, the proposed method use XOR operation to build group key, it's can reduce computation complexity. Besides, the size of transferred data is not influenced by amount of the member. It can improve communication efficiency and reduce data loss rate.

Keyword: Ad Hoc 、 key agreement protocol

誌謝

論文完成後，也代表研究所的學習生活差不多告一個段落了，這二年來，不管在論文的指導上或生活的其它方面，我要感謝的人實在很多，其中最重要的，我要相當感謝我的指導老師羅濟群老師，課業上，羅老師在 Meeting 時總是會給我們許多寶貴的觀念及意見，指引我們正確的思考方向，生活上，羅老師常會帶我們去大自然走走，讓大伙培養健康的身心，真的很感謝羅老師在這二年對我的指導。

此外，我相當感謝博班的俊傑學長，謝謝他在國科會計劃及論文上給我許多的建議跟指導、博班的俊龍學長，謝謝他那麼照顧實驗室的學弟妹，很多生活上的事都可以請他幫忙、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，還有碩一的學弟妹，他們總是帶給 LAB 許多歡樂的氣氛；當然，我還要特別感謝我的家人，尤其是我的父母，要不是他們這麼支持我，我無法這麼幸福地在這裡求學，真的謝謝你們，辛苦了。

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

由於科技的發達，加上許多無線終端設備的普及、盛行，人們可以便利地利用網路來幫助資訊的流通，可以想像一下，你跟一群人走進會議室開會，參與會議的成員們人手一台 PDA 或 Notebook，在這樣的環境裡，不需要事先鋪設實體的網路設線路，也不需要額外的無線基地台(Access Point)，會議成員便可隨即透過 Ad Hoc 無線網路的建構，來做群組通訊，但同時，你又希望會議室外的非法成員無法得知會議成員通訊過程的資訊，即所謂的安全群組通訊環境。

想達成上述的環境，必需借助許多技術，其中密碼學的加解密方法，可以將資料加密後再送給合法的成員，使得非法的成員無法得知加密前的資訊，利用對稱式加密的方法配合資料群播的方式，便可更有效率地將資料傳給其它成員，但此時問題便出現了，所有的群組成員必需擁有同樣的加解密金鑰，才能達成安全且有效率的群組通訊，那麼如何讓群組成員能安全地得到同一支鑰呢？這便是本論文要探討的主題。

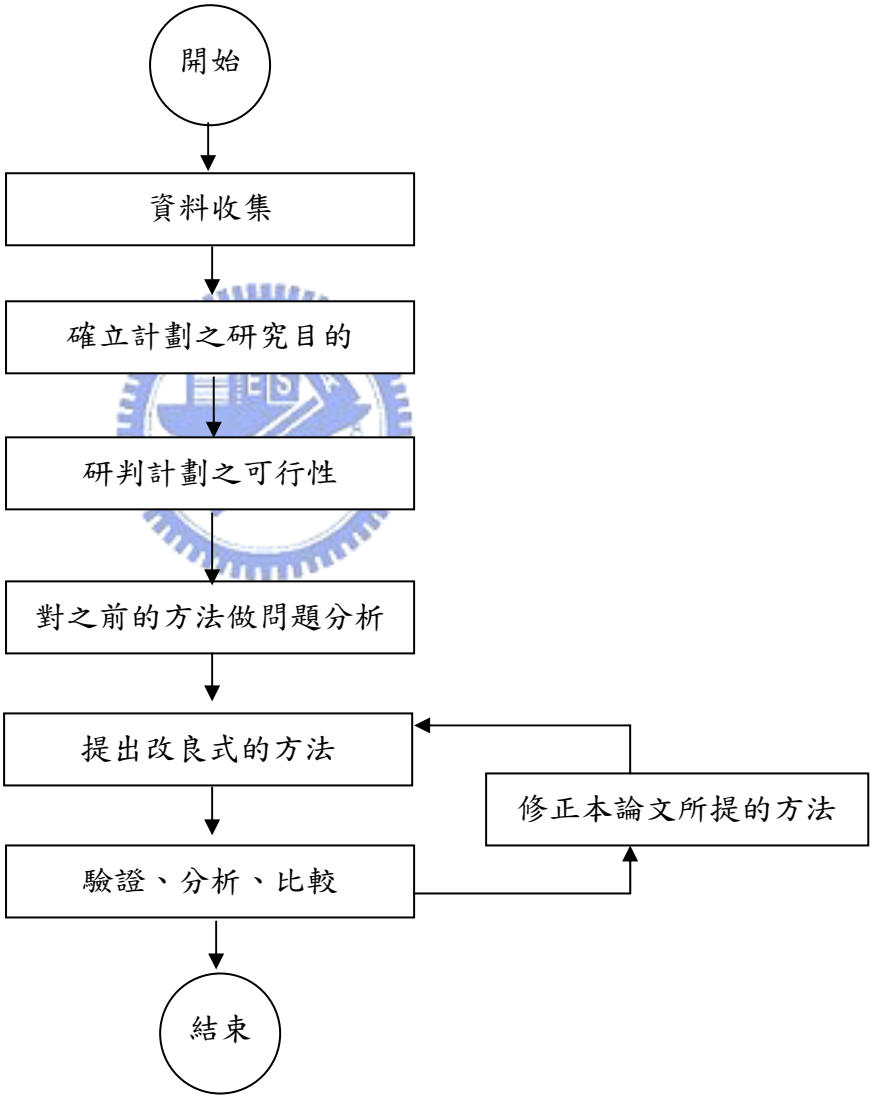
1.2 研究目標

本論文的研究目標是希望於 Ad Hoc 無線網路環境上，提出一個安全、有效率的金鑰協同協定(key agreement protocol)。本論文所提金鑰協定應用的群組通訊環境，是一個封閉式空間，如：會議室・・等，在裡面的網路環境是採用 Ad Hoc 的無線網路，而非一般的實體有線網路或俱基礎建設(infrastructure)的無線網路，目的是希望此空間內的成員能夠利用金鑰協同協定安全地、有效率地產生群組金鑰，以供後續群組通訊時的資料加密使用。

在研究之前學者所提出的方法中發現，大部份的方法都是以 Diffie-Hellman

金鑰交換演算法為基礎，即模指數的運算方法，雖然 Huang 與 Chang 等學者則利用 XOR 的運算方法來改善計算上的效率，但在仔細研究後發現其方法仍有些安全及效率上的問題，因此本論文希望仍以 XOR 方法為基礎，但提出更安全且更有效率的金鑰協同協定。

1.3 研究方法



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：Ad Hoc 無線網路的架構及安全上的挑戰、認證機制的分類及之前學者所提的金鑰協定演算法，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，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・相關研究

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 Ad Hoc 無線網路、認證機制與會議金鑰協定三大部份。

2.1 Ad Hoc 無線網路

由於本論文所提出的金鑰協定主要是用於 Ad Hoc 無線網路的環境，因此本節將對 Ad Hoc 無線網路做簡單及概念性的介紹，包括其架構的介紹與此環境上應注意的安全議題。

2.1.1 架構介紹

一般無線網路依其建構的型態可分成二種，一種為俱有基礎架構(infrastructure)的環境，如:交換機、無線基地台(Access Point)等，其架構如圖 2-1 所示；而另一種則是本論文探討的主題：Ad Hoc 網路，其是一種能夠在沒有事先建置基礎架構的環境下，由各無線端端主機點對點(peer-to-peer)連結所臨時組成的網路，其架構如圖 2-2 所示。Ad Hoc 網路有幾項特色：1.其網路架構俱有動態拓樸的特性，各個連結設備可以處於動態的狀況如位置移動，2.其具有自我組織(self-organization)的能力，可以簡化網路的管理，提高其強健性 (robustness) 和彈性，3.由於它容易且可迅速佈建的特性，Ad Hoc 網路有許多實際的用途，如家庭區域網路、個人區域網路、軍事用途、緊急救災及搜救行動等。

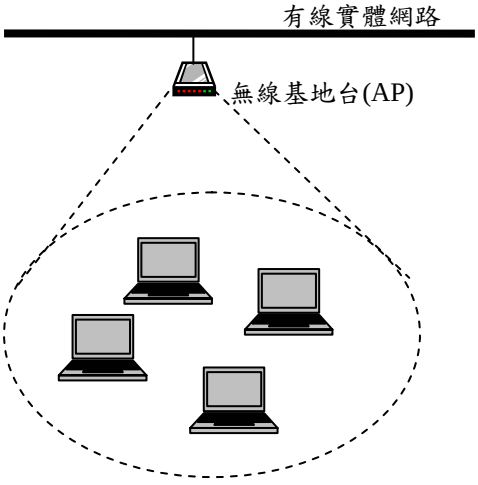


圖 2-1. 俱基礎架構的無線網路

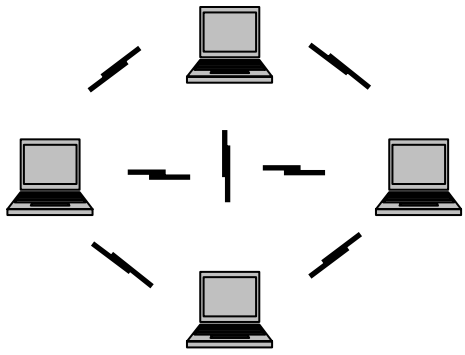


圖 2-2. Ad Hoc 無線網路

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 Ad Hoc 無線網路本身俱有的特性，並非所有的安全協定皆適用於 Ad Hoc 無線網路環境；因此，想要在 Ad Hoc 無線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[5]。

1. 網路攻擊

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以下兩大類：

A. 消極型攻擊 (passive attack)：此類型的攻擊者主要是在網路上竊取、收集別人所傳送的封包資料，希望藉由觀察這些竊取而來的資料可以獲得關於會議金鑰的重要資訊，但他們卻沒有能力去影響合法參與者所傳遞的資料訊息，此類型的攻擊者又稱竊取者(eavesdropper)。

B. 積極型攻擊 (active attack)：此類型的攻擊者有能力影響合法參與者所傳遞的資料訊息，甚至破壞整個會協定的進行，此類型的攻擊常見的方式有：重送攻擊(replay attack)、假冒合法參與者(impersonation)…等方法。

2. 分散式或集中式架構

由於 Ad Hoc 無線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，且沒有一個集中式的裝置。

3. 省電及運算的問題

由於構成 Ad Hoc 無線網路的無線終端設備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、Smart Phone，等，電力及運算能力都有一定的限制，因此；在此環境下所採用的方全運算若是太過複雜，則可能造成設備執行運算上的負擔，及加速電力的

耗損。

2.2 認證機制

想建構一個安全的網路通訊環境，必須有賴許多安全機制的相互配合，而在解決網路安全問題的機制中，使用者身份的驗證是最基本的一步，也是相當重要的一步，因為若系統將非法使用者錯當成合法的使用者，而給予其如同合法使用者般的系統使用權，此系統的安全保護機制便會失去效用。

2.2.1 認證機制分類

通常使用者必需提供一個足以視別自己身份的資訊供服務系統驗證，服務系統才能夠判別該位使用者的合法性，透過使用者身份驗證，系統可以進而決定是否提供服務或控制其存取權限，就目前而言，使用者身份驗證的技術通常可概分為以下三大類[16]：

1. 使用者擁有的 (What a user has)

   例如使用者擁有的 IC 卡(smart card)，通常 IC 卡內會存放一個夠大(128 bits 或 256bits) 且隨機的秘密亂數，用來做為身份認證的安全依據，在認證的過程中，使用者必須出示自己所擁有的實體 IC 卡，配合相關的讀取設備運作以達成視別。

2. 使用者知道的 (What a user knows)

   例如系統事前秘密分配給使用者的通行碼或使用者本身自選的密碼，若是使用者自選的密碼，通常會選擇容易記憶的字串，例如生日數字或熟悉的單字等。

3. 使用者的特徵 (What a user is)

   例如使用者本身的指紋、聲紋、視網膜・・等，即利用生物與生俱來特徵的唯一性來達到身份驗證的目的。

分析以上三類技術，IC 卡雖不易被破解，但使用 IC 卡認證，除了 IC 卡本身的成本之外，還需要相關的讀卡機設備及後續的維護成本，此外，使用者必須

將 IC 卡隨身攜帶也造成使用者的不便，而利用生物特徵的方式也同樣需要額外的辨識設備輔助，而其辨識的正確率及執行效率之間的取捨也使得此類的認證方式還未普遍被採用，相較以上的二種方式，運用使用者所知道的資訊來做識別，如自選通行碼的方式，是目前成本最低、對使用者而言也是非常方便的使用者身份驗證方式。

2.2.2 通行碼驗證金鑰交換協定

在傳統的通行碼身份驗證系統中，密碼的角色只是單純用來檢驗使用者的身份，通常是使用者輸入帳號與密碼，其中密碼經由雜湊函式(hash function)處理後與帳號一併傳送給網路另一端的系統，該系統便可利帳號資訊與其事先已存且已雜湊處理過的密碼資訊做比對，完成使用者身份的驗證；此外，當網路上的某兩方想要建立一個安全的通道以供其倆秘密通訊之用時，最常看到的做法便是兩人共用一把金鑰，做對稱式的資料加密，以保護傳輸資料的安全性。

上述所提關於“密碼認證”與“網路雙方共享金鑰”，兩者原本是彼此獨立的安全機制，但 Bellovin 與 Merritt 於 1992 年[10]首先提出運用於雙方的通行碼驗證交換協定(Password Authentication Key Exchange，PAKE)，一般也稱之為 Encrypted Key Exchange (EKE) 協定，主要目的就是為欲通訊的雙方分配一把金鑰且同時利用通行碼達到互相身份驗證(mutual authentication)，其提出的概念說明如下：

〔EKE 金鑰交換協定的概念〕

假設有 A,B 兩使用者想執行 EKE 協定，其步驟如下：

Step1. 使用者 A 隨機產生一把公開金鑰 E_A，並且利用密碼 P 將其加密後，隨同代號 A 一併送給使用者 B。 〔A→B： $A, P(E_A)$〕

Step2. 使用者 B 隨機產生一隨機值 R，並且利用密碼 P 對使用者 A 送來的資料做解密，解密後得到使用者 A 的公開金鑰 E_A，並且先後用 E_A 與密碼 P 對 R 加密後傳送給使用者 A。 〔B→A： $P(E_A(R))$〕

Step 3. 使用者 A 隨機產生一個挑戰值 $challenge_A$，並且利用自己的私密金鑰(private key)及密碼 P 對使用者 B 送來的資料做解密，解密後得到使用者 B 所產生的 R，再利用 R 對 $challenge_A$ 加密後傳送給使用者 B。 〔A→B： $R(challenge_A)$〕

Step 4. 使用者 B 隨機產生一個挑戰值 $challenge_B$，並且利用自己所握有的隨機值 R 對使用者 A 送來的資料做解密，解密後得到使用者 A 所產生的挑戰值 $challenge_A$，並且再次利用 R 對 $challenge_A$ 及 $challenge_B$ 加密後傳送給使用者 A。 〔B→A： $R(challenge_A, challenge_B)$〕

Step 5. 使用者 A 利用自己所握有的隨機值 R 對使用者 B 送來的資料做解密，解密後得到挑戰值 $challenge_A$ 與 $challenge_B$，並且再次利用 R 對 $challenge_B$ 加密後傳送給使用者 A。 〔A→B： $R(challenge_B)$〕

由以上的步驟說明可看出 EKE 金鑰交換協定的重點在於利用雙方事先已知的密碼 P，來達到身分認證的目的並同時做到金鑰 R 的分配，搭配雙方隨機產生的挑戰值 $challenge_A$、$challenge_B$，可以達到雙方認證(mutual authentication)的效果。

2.3 會議金鑰協定

群組金鑰管理方法可分成三類[9]：1.集中式的金鑰管理：由單一個金鑰分佈中心或管理者來產生群組金鑰 2.半集中式的金鑰管理：將整個群組分成多個子群組，並由各子群組管理者來產生金鑰 3.分散式的金鑰管理：沒有單一個金鑰分佈中心，群組金鑰的建立是每個成員貢獻其密秘值且成員間彼此合作之下而建立的；一般又將前二種類型稱為金鑰分佈協定(key distribution protocol)，而將第三種類型則稱為金鑰協同協定(key agreement protocol)，由於 Ad Hoc 網路環境的特性及為了避免集中式架構下的單一點攻擊的風險，本論文選擇利用分散式的金鑰管理方法，即金鑰協同協定來建立群組金鑰。

由於群組會議中，成員會有加入或離開的動作，所以一般而言，金鑰協同協定除了包含金鑰起始演算法以求出初始金鑰之外，還會包含處理成員加入或離開的演算法，由於本論文的重點在改善初始金鑰演算法，因此以下各節所探討的方法便著重在初始金鑰演算法上。

經由探討後發現，目前大部份的金鑰協同協定(key agreement protocol)都是基於 Diffie-Hellman 金鑰交換的方法而完成，因此以下先說明何謂 Diffie-Hellman 金鑰交換，再介紹相關的金鑰協同協定的研究。

2.3.1 Diffie-Hellman 金鑰交換演算法

Diffie 與 Hellman 兩位學者在 1976 年首先提出公開金鑰密碼系統的觀念[13]，其主要是為了解決一個問題，亦即讓未曾謀面的兩人(two-party)，在不透過第三者的協助下，仍可以透過公眾通道，安全地獲得只有他們倆人才知道的共同金鑰，這就是有名的 Diffie-Hellman 金鑰交換演算法，運作流程如下所示：

〔Diffie-Hellman 金鑰交換演算法〕

參數說明：假設 p 為一個很大的質數，且α為 p 的原根(primitive root)，又 p 與α皆為大家所共知的參數。

Step1. 使用者 i 任選一個隨機亂數 X_i (X_i < P)，計算 $Y_i = \alpha^{x_i} \bmod p$ ，並將Y_i送給使用者 j。

Step2. 同樣地，使用者 j 也任選一個隨機亂數 X_j (X_j < P)，計算 $Y_j = \alpha^{x_j} \bmod p$ ，並將Y_j送給使用者 i。

Step3. 使用者 i 收到Y_j後，計算 $K_i = Y_j^{X_i} \bmod p$ 。

使用者 j 收到Y_i後，計算 $K_j = Y_i^{X_j} \bmod p$ 。

此時，兩人產生的金鑰$K_i = K_j$。

以下將證明使用者 i 計算出的金鑰K_i與使用者 j 計算出的金鑰K_j是真的是相等的：

$$
\begin{aligned}
K_i &= Y_j^{x_i} \bmod p \\
&= (\alpha^{x_j} \bmod p)^{x_i} \bmod p \\
&= (\alpha^{x_j})^{x_i} \bmod p \\
&= (\alpha^{x_i})^{x_j} \bmod p \\
&= (\alpha^{x_i} \bmod p)^{x_j} \bmod p \\
&= Y_i^{x_j} \bmod p = K_j
\end{aligned}
$$

Diffie-Hellman 金鑰交換演算法的安全性取決於解離散對數問題(Discrete Logarithm Problem)之困難度，因為當 p 大到一定程度時，欲由 y 求出 x，為計算上不可能；基於 Diffie-Hellman 金鑰交換演算法，許多會議金鑰協定陸續被提出，以下各小節將陸續做介紹。

2.3.2 Group Diffie-Hellman 金鑰協同協定

由於基本的 Diffie-Hellman 金鑰交換演算法只拘限於兩個人之間(two-party)的運作，所以 Steiner、Tsudik、Waidner 三位作者於 1996 年，提出 Group Diffie-Hellman 金鑰分配演算法[6]，使得多個人之間(n-party)也能建立共享金鑰，以供後續安全的群組通訊使用，其提出的演算法有 GDH.1、GDH.2、GDH.3 三種版本，GDH.2 主要是改善 GDH.1 在成員合作回合數、傳輸訊息量等效率問題，而 GDH.3 則是為了改善前二版在指數運算效率的問題，但在成員合作回合數及傳輸訊息量則均較 GDH.1 與 GDH.2 差，綜合所探討的文獻，大部份都是以 GDH.2 為比較對象，因此本節中介紹的是 GDH.2 的演算法，其演算法如下所示：

〔GDH.2 金鑰協定演算法〕

參數說明：p 與 q 皆為質數，且 $q = 2p + 1$，α 為 Z_q^*(註 1)下唯一循環子群組的生成數(generator)。

回合 i (1 ≤ i ≤ n-1)：

Step 1. 成員 M_i 選擇一個隨機數 r_i ， $r_i \in Z_q^*$

Step 2. 成員 M_i 傳給 M_{i+1}：$\alpha^{\prod_{k=1}^{i} r_k}$ 、$\alpha^{(\prod_{k=1}^{i} r_k)/r_j}, \forall 1 \le j \le n$

回合 n：

Step 1. 成員 M_n 選擇一個隨機數 r_n ， $r_n \in Z_q^*$

Step 2. 成員 M_n 廣播(broadcast)給每個成員 M_i：$\alpha^{\prod_{k=1}^{i} r_k}$ 、$\alpha^{(\prod_{k=1}^{i} r_k)/r_j}, \forall 1 \le j \le n$

Step 3. 每一個成員 M_i 計算出最後的群組金鑰 $K_i = \alpha^{\prod_{j=1}^{n} r_j}$

註 1：Z_q^* 為數值 1 到 q-1 中，與 q 互質的數所形成的集合。

在 GDH.2 金鑰協定中，其成員合作的傳輸模式是採環狀的架構，以四個成員合作為例子來說明此演算法的執行過程，如圖 2-3，M_1 挑選一隨機數 r_1，並利用模指數運算計算出 $Y_1 = \alpha^{r_1} \bmod p$ 後，將 Y_1 傳送給 M_2，M_2 也挑選一隨機數 r_2，計算出 $Y_2 = \alpha^{r_2} \bmod p$ 後，再利用 r_2 與 Y_1 進行 Diffie-Hellman 金鑰運算後產生 $Y_{12} = \alpha^{r_{12}} \bmod p$，並將 Y_1, Y_2, Y_3 送給 M_3，同理，M_3 可以計算出 $Y_{12}, Y_{13}, Y_{23}, Y_{123}$（即 $\{\alpha^{r_{12}}, \alpha^{r_{13}}, \alpha^{r_{23}}, \alpha^{r_{123}}\} \bmod p$）後送給 M_4，在 M_4 挑選一隨機數 r_4 後，其便可以算出最後的會議金鑰 $K = \alpha^{r_{1234}} \bmod p$，$M_4$ 還必須將其計算產生後的 $\{\alpha^{r_{234}}, \alpha^{r_{134}}, \alpha^{r_{124}}\} \bmod p$ 利用廣播傳送給 M_1、M_2、M_3，最後，M_1、M_2、M_3 便能利用本身握有的隨機值 r_i 與所收到的廣播資料，透過 Diffie-Hellman 金鑰運算產生會議金鑰 K。

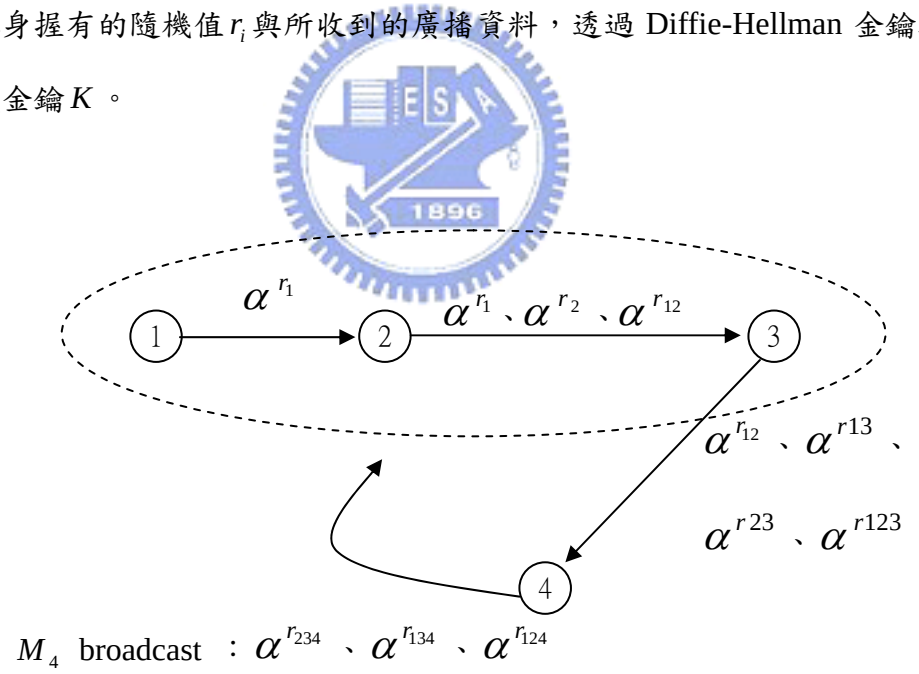


圖 2-3. 四人合作的 GDH.2 金鑰協定

2.3.3 Hypercube 金鑰協同協定

此協定演算法是由 Becker 與 Wille 於 1998 年所提出的[4]，同樣是將 Diffie-Hellman 金鑰交換演算法擴充至多人環境使用，以求多人之間(n-party)建立共享金鑰，其成員合作關係是採用向量空間的立方體概念，目的是想改善群體成員建立金鑰過程之合作回合數的效率問題，其演算法如下所示：

〔Hypercube 金鑰協同協定演算法〕

參數說明：假設成員共有 2^d 個人，成員合作關係為 d 因次向量空間 $GF(2)^d$ (註 1)，

$\vec{b}_1,...,\vec{b}_d$ 為 $GF(2)^d$ 的基準方向，q 為質數，α 為 Z_q^*(註 2)下循環子群組的生成數(generator)。

回合 1：

Step1. 每一個成員 $\vec{v} \in GF(2)^d$ 選擇一個隨機數 $r_{\vec{v}}$ ， $r_{\vec{v}} \in Z_q^*$ 。

Step2. 每一個成員 $\vec{v}$ 與成員 $\vec{v}+\vec{b}_1$ 分別利用其秘密值 $r_{\vec{v}}$ 、 $r_{\vec{v}+\vec{b}_1}$ 進行 Diffie-Hellman 金鑰交換。

回合 i ($1 < i \le d$)：

Step1. 每一個成員 $\vec{v}$ 與成員 $\vec{v}+\vec{b}_1$ 進行 Diffie-Hellman 金鑰交換，雙方所使用的秘密值皆為回合 i-1 中 Diffie-Hellman 金鑰交換後的值。

當合回 i 結束後，每位成員皆已產生相同的會議金鑰。

註 1：$GF(\ )$ 為 Galois Field ，又叫有限場。

註 2：Z_q^* 為數值 1 到 q-1 中，與 q 互質的數所形成的集合。

以四個成員為例子來說明此演算法的執行過程，如圖 2-4 與圖 2-5 所示，其成員合作關係為2^2 Cube；在第一回合時，每個成員各自產生一個秘密隨機值S_i，成員 A 與成員 B 各自利用其秘密值S_A、S_B合作進行 Diffie-Hellman 金鑰交換後產生子金鑰$S_{AB}=\alpha^{S_AS_B}$，同理，成員 C 與成員 D 也利用其秘密值進行 Diffie-Hellman 金鑰交換產生子金鑰$S_{CD}=\alpha^{S_CS_D}$，在第二回中，成員 A 改為和成員 C 合作，分別利用其前一回合(第一回合)合作後所產生的子金鑰S_{AB}、S_{CD}進行 Diffie-Hellman 金鑰交換，產生會議金鑰$K=\alpha^{S_{AB}S_{CD}}$，同理，成員 B 也改為和成員 D 合作，利用 Diffie-Hellman 金鑰交換產生會議金鑰$K=\alpha^{S_{AB}S_{CD}}$。

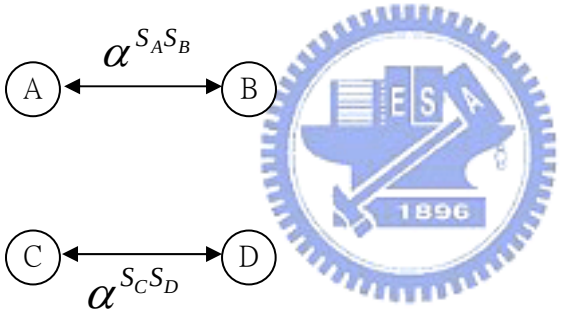


圖 2-4. 四成員合作的第一回合

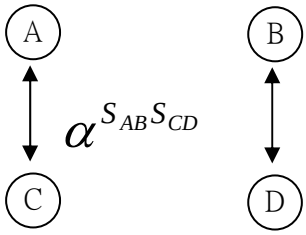


圖 2-5. 四成員合作的第二回合

Hypercube 演算法有一個重要限制，即其參與成員數目必需符合$2^d, d>1$才能進行，關於此缺點，[4]中所提的 Octopus 演算法便是以 Hypercube 演算法為基礎，加上子群組的概念，來改善參與成員數目限制的問題。

2.3.4 DH-LKH 金鑰協同協定

Wallner 等學者於 1997 年提出利用階層式樹狀結構(Logical Key Hierarchy)[1][2]來做金鑰管理，藉由一個金鑰分配中心(KDC)來產生、維護金鑰；而 Kim 等學者在 2000 年所提出的 DH-LKH(又稱為 T-GDH)金鑰協同協定，則是利用階層式樹狀結構的概念加上 Diffie-Hellman 金鑰交換演算法所構成，主要目的是讓會議金鑰的建立改由所有成員的貢獻值組成，且藉由樹狀架構來增加金鑰更新的效率，此演算法架構說明如下：

〔DH-LKH 演算法架構說明〕

1. 假設所有成員依其編號形成一個平衡的二元樹狀關係，每個葉節點代表一個成員。
2. 此樹狀架構的根節點階層為 0，最淺階層則為 h，每每節點可以用編號$<l,v>$表示，代表該節點位於第 l 階層中的第 v 個位置節點，且$(0 \le v \le 2^l - 1)$。
3. 某個節點編號$<l,v>$若擁有左右子節點，則左右子節點編號各為$<l+1,2v>$、$<l+1,2v+1>$。
4. 每一個節點皆有二個金鑰，分別為秘密金鑰 $K_{<l,v>}$、隱藏金鑰 $BK_{<l,v>} = f(K_{<l,v>})$；

   $f(\ )$ 方程式如同 Diffie-Hellman 金鑰演算法中的模指數運算，$f(k) = \alpha^k \bmod p$。
5. 每個成員 i 知道樹狀結構中的所有 $BK_{<l,v>}$，該集合用 BK^*_i 表示。
6. 位於$<l,v>$的成員 i 知道從$<l,v>$到$<0,0>$路徑中的所有 $K_{<l,v>}$，該集合用 KEY^*_i 表示。
7. 形成 KEY^*_i 路徑上每個節點的兄弟節點所形成的集合用 CO_i 表示。
8. 每把秘密金鑰

$$
\begin{aligned}
K_{<l,v>} &= (BK_{<l+1,2v+1>})^{K_{<l+1,2v>}} \bmod p \\
&= (BK_{<l+1,2v>})^{K_{<l+1,2v+1>}} \bmod p \\
&= \alpha^{K_{<l+1,2v>} K_{<l+1,2v>}}
\end{aligned}
$$

以六個成員為例，如圖 2-6，當成員 M_5 要計算出會議金鑰時，當其知道 $BK_5^* = \{BK_{<0,0>}, BK_{<1,1>}, ..., BK_{<3,7>}\}$ 和 $KEY_5^* = \{K_{<3,6>}, K_{<2,3>}, K_{<1,1>}, K_{<0,0>}\}$ 後，M_5 可計算出會議金鑰 $K = \alpha^{(\alpha^{r_4(\alpha^{r_5 r_6})})(\alpha^{r_3(\alpha^{r_1 r_2})})}$ 。

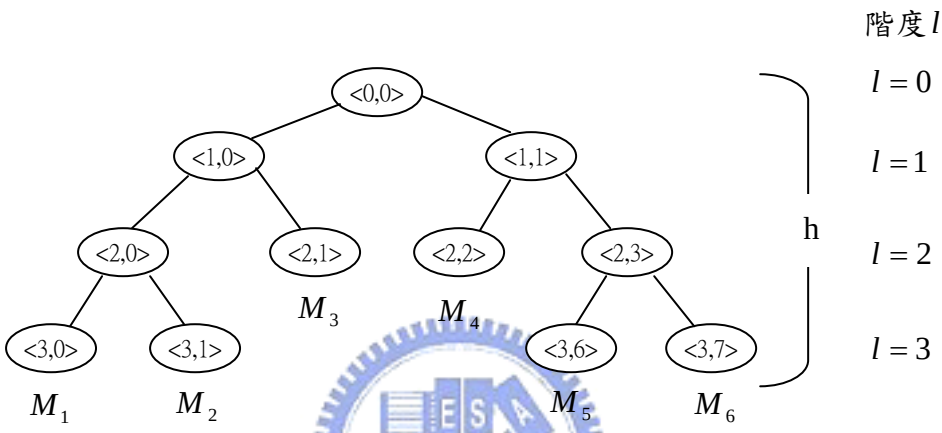


圖 2-6. 6 個成員的 DH-LKH 架構

以階層式樹狀為結構為基礎的金鑰協同協定除了 DH-LKH 之外，還有分散式階層樹狀結構(Distributed LKH) [7]、分散式單向函式樹(Distributed One-Way Function Tree) [12] 等方法。

2.3.5 Huang 與 Chang 所提的金鑰協同協定

2003 年由 Huang 與 Chang 兩人所提出的金鑰協同協定[8]，與基於 Diffie-Hellman 金鑰交換方法最大的不同在於此協定完全沒用到 Diffie-Hellman 演算法來求出共同金鑰，而是利用互斥或(XOR)運算式，可避開採用 Diffie-Hellman 方法所產生的高度模指數運算，因而更適合運算能力較差的行動通訊裝置，此外，此協定亦加上了密碼認證來驗證使用者身份的合法性。

在此演算法裡，由成員所貢獻的秘密值，其合作的傳輸模式是採用樹狀結構，而且是一個完全二元樹的結構，如圖 2-7，其中編號最大的(M_n)為檢查者(Checker)，其地位和其它成員最大的不同之處，在於除了也得貢獻組成金鑰的秘密值之外，還要檢查最後每位成員經由合作所握有的金鑰是否一致，而編號第二大的(M_{n-1})為候補者(Candidae)，負責遞補其它成員離開時的編號，讓成員們的合作關係保持在完全二元樹的結構上，每位成員皆會貢獻自己所產生的秘密值，而最後的共同金鑰則是由所有成員所貢獻的秘密值相互執行 XOR 運算而成。

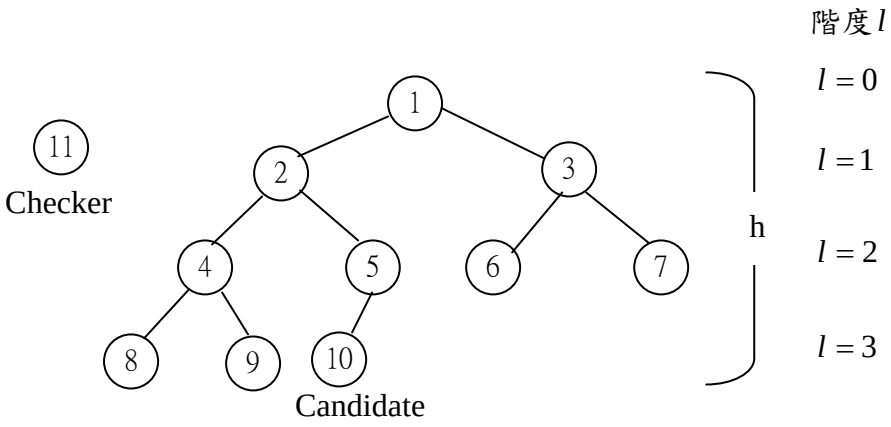


圖 2-7. 依成員編號所形成的虛擬樹狀結構

■ 演算法

Huang & Chang 會議金鑰演算法之金鑰起始演算法分為二個階段，說明如下：

表 2-1. Huang & Chang 演算法中的符號定義

符號	說明
M_i	編號為 i 的成員
P	事先已知的身份認證用密碼
$f(\ )$	單向雜湊函式
S_i 、S_i'	成員 M_i 所產生的隨機秘密值
K_i'	成員 M_i 所計算出的子金鑰
π	成員 M_1 所計算出的子金鑰
$\oplus$	XOR 運算
$\|\|$	串接
E_x	利用 x 為金鑰做對稱式加密

〔Huang & Chang 金鑰起始演算法之第一階段〕

成員們的合作順序是由樹的最高階(階度= h)往最低階(階度 = 0)，對每一個階度而言，依成員編號的不同而有不同的處理：

CASE 1 (2i > n-1)： M_i 為葉節點

$$K_i' = S_i$$

CASE 2 (2i < n-1)： M_i 有左右兩子節點

$$K_i' = K_{2i}' \oplus K_{2i+1}' \oplus S_i$$

CASE 3 (2i = n-1)： M_i 只有左子節點

$$K_i' = K_{2i}' \oplus S_i$$

CASE 4 (i = 1)： M_i 為根節點

$$K_1' = K_2' \oplus K_3' \oplus S_1 = \pi$$

每一個員利用 $f(p \| K_i')$ 做為身份認證依據，並與 K_i' 一同傳給父節點。

〔Huang & Chang 金鑰起始演算法之第二階段〕

Step 1. $M_1 \xrightarrow{Broadcast} M_{i+1}$; i = 1 …n-2

$\pi, f(P \| \pi)$

Step 2. $M_i \longrightarrow M_n$; i =1…n-1

$C_i, f(P \| C_i)$ 註： $C_i = \pi \oplus S_i \oplus S_i'$

Step 3. $M_n \xrightarrow{Broadcast} M_i$; i =1…n-1

$E_{p \oplus C_i}(C_i \oplus S_n)$

Step 4. $M_i \longrightarrow M_n$; i =1…n-1

$E_{p \oplus S_n}(K_i)$ 註： $K_i = \pi \oplus S_n$

Step 5. M_n 檢查每位成員的會議金鑰是否相同

以 11 個成員為例子說明此演算法的執行過程，在第一階段演算法中，每一個葉節點成員 M_i $(i = 6,7,8,9,10)$各自選擇一個隨機數值 S_i 且讓 $K_i' = S_i$，並為 K_i' 產生一個驗證值 $F_i = f(P \| K_i')$ 後，將 K_i' 與 F_i 送到它的父節點成員；父節點 M_i ($i = 3,4,5$) 收到子節點送來的資料後，先利用 $F_{2i} = f(P \| K_{2i}')$ 、$F_{2i+1} = f(P \| K_{2i+1}')$ 來驗證 K_{2i}' 與 K_{2i+1}' 的合法性，若合法，則選擇一個隨機數值 S_i 並計算 $K_i' = K_{2i}' \oplus K_{2i+1}' \oplus S_i$，並且為 K_i' 產生一個驗證值 $F_i = f(P \| K_i')$；再將 K_i' 與 F_i 送到它的父節點成員；同樣的程序，一直進行到 M_1，此時 M_1 所產生的 $K_i' = K_{2i}' \oplus K_{2i+1}' \oplus S_i$ 等同於由 n-1 個成員所合作產生的子金鑰 $\pi = K_1' = K_2' \oplus K_3' \oplus S_1$，$M_1$ 再為 π 產生一個驗證值 $F_1 = f(P \| \pi)$，完成第一階段演算法，結果如圖 2-8 所示。

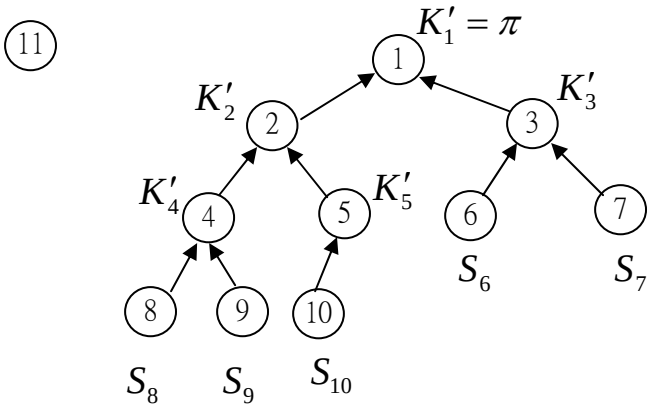


圖 2-8. Huang & Chang 方法之第一階段結果

說明：

$$K_4' = S_4 \oplus S_8 \oplus S_9$$

$$K_5' = S_5 \oplus S_{10}$$

$$K_2' = S_2 \oplus K_4' \oplus K_5'$$

$$K_3' = S_3 \oplus S_6 \oplus S_7$$

$$K_1' = S_1 \oplus K_4' \oplus K_5' = S_1 \oplus S_2 \oplus S_3 \oplus ... \oplus S_9 \oplus S_{10} = \pi$$

在第二階段演算法中，成員 M_1 利用廣播(broadcast)的方式將 π 、$F_i = f(P \| \pi)$ 送給其它成員 M_i(i = 2,3,…,11)；成員 M_i(i = 1,2,…,10)收到由 M_1 送來的資料後，再次選擇一隨機數 S_i' ，計算 $C_i = \pi \oplus S_i \oplus S_i'$ ，並將 C_i 與 $F_i = f(P \| C_i)$ 傳給成員 M_{11}；成員 M_{11} 收到其它成員送來的資料後，選擇一隨機數 S_{11} ，利用 $P \oplus C_i$ 當金鑰對 $C_i \oplus S_{11}$ 做對稱式加密，並將 $E_{p \oplus C_i}(C_i \oplus S_{11})$ 送給各

個成員 M_i (i = 1,2,…,10)；成員 M_i (i = 1,2,…,10)收到由 M_{11} 送來的資料後，執行解密 $D_{p\oplus C_i}(C_i \oplus S_{11})$，並計算出最終的會議金鑰 $K_i = \pi \oplus S_{11}$，再將 $E_{p\oplus S_{11}}(K_i)$ 送給成員 M_{11}；最後，成員 M_{11} 檢查其它成員所送來的會議金鑰是否一致，完成第二階段演算法，結果如圖 2-9 所示。。

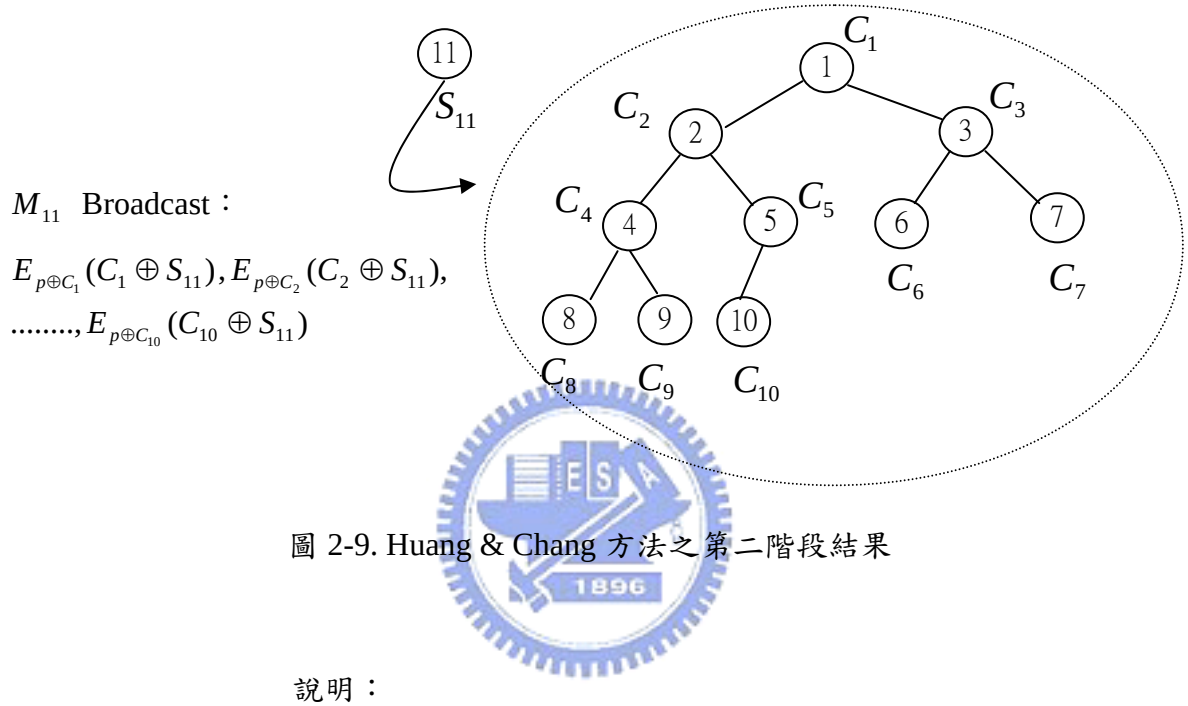


圖 2-9. Huang & Chang 方法之第二階段結果

說明：

會議金鑰 $K = \pi \oplus S_{11} = S_1 \oplus S_2 \oplus S_3 \oplus ... \oplus S_{10} \oplus S_{11}$

三、改良後的金鑰協同協定

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 Ad Hoc 的網路環境上較適用金鑰協同協定，而不採用有 CA 的架構的金鑰協定，也了解利用 XOR 運算的好處，接著本論文提出一個改良後的金鑰協同協定，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 Ad Hoc 應用於無線網路環境上的設備，其電力或運算能力常被假設有一定的限制，綜觀之前學者所提的方法，利用 XOR 運算取代模指數的的確可以增加運算上的效率[11]，因此本論文希望所提出的方法也是以 XOR 運算為基礎的；此外，無線網路的封包較一般實體網路更容易被擷取、收集，因此不管是主動式攻擊或被動式攻擊對於無線網路安全的威脅都相當大，以是否能承受網路上非法的惡意攻擊來檢視於第二章中所探討過的金鑰協定，在仔細檢視於第二章所述的 Huang 與 Chang 方法後，發現此方法很有可能遭受到某些非法的惡意攻擊，而違害到整個金鑰建置的安全，以下便是其可能遭受攻擊的探討與驗證：

首先說明在一個 Ad Hoc 無線網路的環境下，配合 Huang 與 Chang 方法做 Key Agreement 時，假設一個非法的惡意攻擊者有能力擷取到此協定過程的所有封包，則其可以擷取到的封包資料列於表 3-1 中：

表 3-1. 運用 Huang & Chang 方法時，攻擊者可以擷取到的封包

資料	索引值
$f(P \parallel K_i')$	i = 2,3,…,n-1
K_i'	i = 2,3,…,n-1
$f(P \parallel \pi)$	
$\pi = S_1 \oplus S_2 \oplus S_3 \oplus \ldots\ldots\ldots\ldots \oplus S_{n-1}$	
$C_i = \pi \oplus S_i \oplus S_i'$	i = 1,2,…,n-1
$f(P \parallel C_i)$	i = 1,2,…,n-1
$E_{p \oplus C_i}(C_i \oplus S_n)$	i = 1,2,…,n-1
$E_{p \oplus S_n}(\pi \oplus S_n)$	i = 1,2,…,n-1
註：n = 成員數目	

以下說明此協定所會遭遇的問題：

✓ 重送攻擊 (Replay Attack)

由於攻擊者可以輕易收集到多組 $f(P \parallel K_i')$ 與 K_i' 的配對，雖然攻擊者並不知道密碼 P 為何，但只要將收集到的一組 $f(P \parallel K_i')$ 與 K_i' 配對，往某一個攻擊的目標成員送，並謊稱自己是該成員的子節點成員，雖然該目標成員會做身份驗證的動作，但仍會驗證合格，即信以為真的接受攻擊者所重送的資料，造成金鑰建立過程的混亂與錯誤發生。

舉例說明：

如圖 3-1 所示，可以觀察出，假若有一個非法的惡意的攻擊者 A 擷取到成員 4 送給成員 2 的封包資料 $f(P \| K_4')$、K_4'，此攻擊者可以謊稱自己是成員 4 或 5，而將此封包資料重送給成員 2；甚至謊稱自己是合法的成員 2，而將 $f(P \| K_4')$ 、K_4'送給成員 1，雖然攻擊者並不知道驗證身分的密碼P為何，但成員 1 利用雜湊函式做合法性驗證後，仍會誤信攻擊者是合法的成員 2，使得金鑰建立的過程產生錯誤。

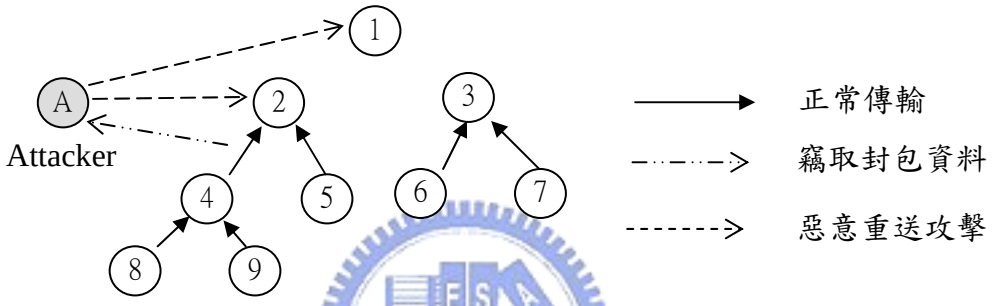


圖 3-1. 重送攻擊示意圖

✓ 密碼猜測攻擊 (Password Guessing Attack)

One-Way Hash Function 的定理指出，若 F( ) 是一個 One-Way Hash Function 且 F(X) = Y，則給一個 X，可以很快求出一個對應的 Y，但給一個 Y，並無法求出相對應的 X；在此協定中，攻擊者可以收集到多組執行完One-Way Hash Function後的值，如：$f(P \| K_i')$、$f(P \| \pi)$ 、 $f(P \| C_i)$，但其中的K_i'、π 、C_i，由於是以明文傳送，所以惡意的攻擊者可輕易得到，又假設該惡意攻擊者也知道此演算法所運用的 One-Way Hash Function，此時問題便出現了，由於人類一次的記憶長度有限，所以供人們記憶的密碼通常不會太長，頂多 8 到 10 個文數字，此時，攻擊者便可利用某個配對，如：

$f(P \| K_i')$與K_i'，加上字典猜測或暴利法猜測來執行密碼猜測攻擊，一旦此協定的密碼被猜出來，則攻擊者便可利用其收集到的資料：$E_{p \oplus C_i}(C_i \oplus S_n)$、$C_i$、$\pi$ ，計算出此次的會議金鑰。

✓ 執行效率上的問題

除了上述二個安全上的問題之外，還有執行效率上的問題，觀察其演算法之第二階段中發現，若總共有 n 個成員參與這個協定，則成員M_n必須要將其自己產生的秘密值S_n，利用$P \oplus C_i$當加密金鑰做(n-1)次加密，產生$E_{p \oplus C_i}(C_i \oplus S_n)$; for i = 1,2,…,n-1；並且將這(n-1)個加密後的資料利用廣播(broadcastt)一併傳送給各個成員$(M_1, M_2, \ldots, M_{n-1})$，很明顯的，當參與會議的成員一多，則金鑰建立的時間便會卡在M_n做加密的時間，而影響到整體的效能，且M_n所廣播的資料量大小，也會隨著成員數而增加。

3.2 改良後之金鑰協同協定

在前一小節裡，做完問題分析之後，本論文所要做的方向，便是改良 Huang 與 Chang 的方法，即基於 Huang 與 Chang 的演算法架構，但要能解決 3.1 節所分析出來的問題；本論文同樣是利用 XOR 的運算來產生最後的會議金鑰，整個過程並不會用到任何 Diffie-Hellman 模指數運算，但在成員合作所傳輸的資料封包與傳輸流程做了些改變，以避免重送攻擊、密碼猜測攻擊，並且克服之前的協定運作效率上的問題；因此，在本論文所提的方法裡，除了改良產生初始會議金鑰的演算法之外，還增加了會議金鑰更新的演算法，用於持續一段時間沒有成員加入或進開會議的情形之下，讓金鑰安全強度得以提高，此小節便是對本論文所提的金鑰協定做詳細的介紹。

3.2.1 成員合作的傳輸架構

由於本論文所提的方法是將 Huang 與 Chang 的方法做改良，所以改良後的金鑰協同協定其成員合作傳輸架構仍是以 Huang 與 Chang 方法所提的架構為基礎，在第二章中已做過介紹，如圖 2-7 所示。

3.2.2 演算法中的符號定義

關於本論文所提之演算法中的符號定義，說明如下表：

表 3-2. 本論文所改良演算法中的符號定義

符號	說明
M_i	編號為 i 的成員
M_c	指該成員的角色為子節點(child)
M_p	指該成員的角色為父節點(parent)
ID_i	成員 i 的 ID 編號
P	身份認證用的密碼
$f(\ )$	單向雜湊函式
S_i、K_i'、K_i、π	長度與會議金鑰長度相同的數
$nonce_i$	成員 i 所產生的隨機亂數
$\oplus$	XOR 運算
$\|\|$	串接
E_x	利用 x 為金鑰做對稱式加密
D_x	利用 x 為金鑰做對稱式解密

3.2.3 演算法

本論文所提的演算法有二： 1.產生初始會議金鑰的演算法 2.會議金鑰更新的演算法，此二種演算法的作用是不同的，詳細說明如下：

一．產生初始會議金鑰的演算法

本論文所提的初始金鑰演算法用於產生初始會議金鑰，此演算法同樣分成二個階段，第一個階段是讓 n -1 個成員合作產生子金鑰，第二個階段則是讓 n 個成員合作產生會議金鑰，演算法內容說明如下：

〔本論文所提金鑰起始演算法之第一階段：n -1 個成員合作產生子金鑰〕

成員們的合作順序是由樹的最高階(階度= h)往最低階(階度 = 0)，對每一個階度而言，依成員編號及角色的不同而有不同的處理：

若 $M_c = M_i$ ，則 $M_p = M_{\lfloor i/2 \rfloor}$

若 $ID_c = ID_i$ ，則 $ID_p = ID_{\lfloor i/2 \rfloor}$

CASE 1 ($i \neq n$)

Step1. $M_c \longrightarrow M_p$ ： ID_c 、 $E_p(ID_c \| nonce_c)$

Step2. $M_c \longleftarrow M_p$ ： ID_p 、 $E_p(ID_p \| nonce_c \| nonce_p)$

Step3. $M_c \longrightarrow M_p$ ： ID_c 、 $E_p(ID_c \| nonce_p \| K_i')$

若(2i > n-1)， M_i 為葉節點：

$K_i' = S_i$

若(2i < n-1)， M_i 為有左右子節點的父節點：

$K_i' = S_i \oplus K_{2i}' \oplus K_{2i+1}'$

若(2i = n-1)， M_i 為有左子節點的父節點：

$K_i' = S_i \oplus K_{2i}'$

CASE 2 (i = 1)：

$\pi = S_1 \oplus K_{2i}' \oplus K_{2i+1}' = S_1 \oplus S_2 \oplus ..... \oplus S_{n-1}$

〔本論文所提金鑰起始演算法之第二階段：n 個成員合作產生會議金鑰〕

Step1. $M_1 \xrightarrow{Broadcast} M_i$ ； (i = 2,3,….n)：

ID_1 、 $E_p(ID_1 \| \pi \| nonce_1)$

Step2. $M_n \xrightarrow{Broadcast} M_i$ ； (i = 1,2,…,n-1)：

ID_n 、 $E_{f_{(p\|\pi)}}(S_n \| nonce_n)$

Step3. $M_i \longrightarrow M_n$ ； (i =1,2,…n-1)：

ID_i 、 $f(K_i)$

Step4. M_n 檢查每位成員的會議金鑰是否相同

為了和Huang & Chang的方法做比較，如同第三章所舉之例子，同樣以 11 個成員為例子說明此演算法的執行過程，<u>在第一階段演算法中</u>：

(1).每一個葉節點成員 M_i $(i = 6,7,8,9,10)$各自選擇一個隨機數值 $nonce_i$ ，並利用事先已知的密碼 P 做為對稱式加密的金鑰對 $(ID_i \| nonce_i)$ 加密產生 $E_p(ID_i \| nonce_i)$ ，將 ID_i 與 $E_p(ID_i \| nonce_i)$ 送到它的父節點成員。

(2).父節點 M_j $(j = 3,4,5)$收到子節點送來的資料後，利用密碼 P 解出子節點的 ID 與 $nonce$ 來驗證子節點的合法性，若合法，則選擇一個隨機數值 $nonce_j$ 並計算 $E_p(ID_j \| nonce_i \| nonce_j)$ ，並將 ID_j 、 $E_p(ID_j \| nonce_i \| nonce_j)$ 反送到它的

子節點。

(3).每一個子節點成員 M_i $(i = 6,7,8,9,10)$收到父節點 M_j 送來的資料後，再利用密碼 P 解出父節點的隨機數值 $nonce_j$，並且再次選擇一個隨機數值(貢獻值) S_i 且讓 $K_i' = S_i$，計算出 $E_p(ID_i \| nonce_i \| K_i')$ 後，再將 ID_i、$E_p(ID_i \| nonce_i \| K_i')$ 送到它的父節點成員，完成以上三方交握的流程之後，父節點成員便可以安全地獲得子葉節點的貢獻值。

(4).同樣的程序，若該節點是內部節點(如：M_2, M_3, M_4, M_5)但有父節點，則其送給父節點的貢獻值是自己所產生的隨機值與子節點貢獻值 XOR 運算後的結果 $K_i' = S_i \oplus K_{2i}' \oplus K_{2i+1}'$，當一直進行到 M_1，此時 M_1 所產生的 $K_i' = K_{2i}' \oplus K_{2i+1}' \oplus S_i$ 等同於由 n-1 個成員所合作產生的子金鑰 $\pi = K_1' = K_2' \oplus K_3' \oplus S_1$，完成第一階段演算法，結果如圖 3-2 所示。

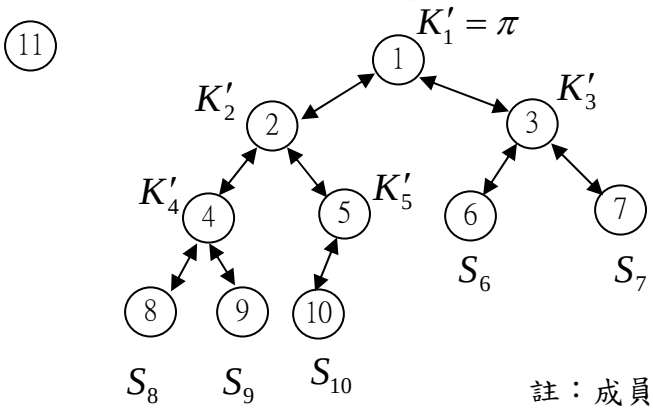


圖 3-2. 本論文所提初始金鑰演算法之第一階段結果

說明：

$K_4' = S_4 \oplus S_8 \oplus S_9$

$K_5' = S_5 \oplus S_{10}$

$K_2' = S_2 \oplus K_4' \oplus K_5'$

$K_3' = S_3 \oplus S_6 \oplus S_7$

$K_1' = S_1 \oplus K_4' \oplus K_5' = S_1 \oplus S_2 \oplus S_3 \oplus ... \oplus S_9 \oplus S_{10} = \pi$

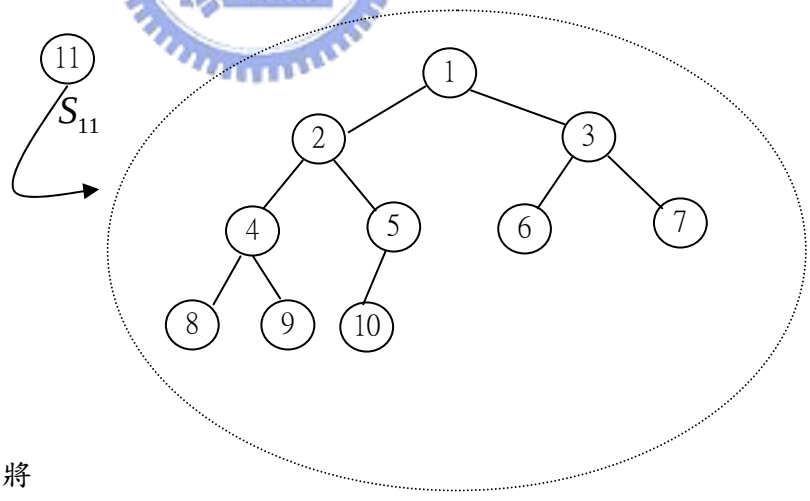
在第二階段演算法中：

(1).成員 M_1 產生一隨機數值 $nonce_1$，並計算 $E_p(ID_1 \| \pi \| nonce_1)$ 後，利用廣播(broadcast)的方式將 ID_1、$E_p(ID_1 \| \pi \| nonce_1)$ 送給其它成員 M_i (i = 2,3,…,11)。

(2).成員 M_{11} 收到成員 M_1 送來的資料後，選擇二個隨機數 S_{11} (貢獻值)、$nocne_{11}$，並利用 $f(P \| \pi)$ 做為對稱式加密法的金鑰對 $(S_{11} \| nonce_{11})$ 加密後，將 ID_{11}、$E_{f_{(p\|\pi)}}(S_n \| nonce_n)$ 利用廣播送給各個成員 M_i (i = 1,2,…,10)。

(3).成員 M_i (i = 1,2,…,10)收到由 M_{11} 送來的資料後，執行解密得到 S_{11}，並計算出最終的會議金鑰 $K_i = \pi \oplus S_{11}$，再將 ID_i、$f(K_i)$ 送給成員 M_{11}。

(4).最後，成員 M_{11} 檢查其它成員所送來的會議金鑰是否一致，完成第二階段演算法，結果如圖 3-3 所示。。



M_{11} Broadcast：

$E_{f(p\|\pi)}(S_{11} \| nonce_{11})$

註：M_n 只要加密一次即可將資料送出

圖 3-3. 本論文所提初始金鑰演算法之第二階段結果

說明：

會議金鑰 $K = \pi \oplus S_{11} = S_1 \oplus S_2 \oplus S_3 \oplus ... \oplus S_{10} \oplus S_{11}$

二．更新金鑰的演算法

在初始金鑰產生之後，群組通訊便得以安全地開始進行，但假使持續一段時間沒有成員加入或進開會議環境時，會議金鑰並不會變動，為了提高會議金鑰的安全強度，本論文提出用於此環境下的更新金鑰演算法，說明如下：

〔本論文所提更新金鑰演算法：在沒有成員離開或加入的情況下更新金鑰〕

Step1. $M_1 \xrightarrow{Broadcast} M_i$ ； (i = 2,3,….n)：

ID_1 、 $E_p(ID_1 \| S_1'' \| nonce_1)$

Step2. M_i 計算新的會議金鑰 $K_{new} = K_{old} \oplus S_1''$ ； (i = 1,2,…,n)

Step3. $M_i \longrightarrow M_1$ ； (i =2,3,…n)：

ID_i 、 $f(K_{new})$

Step4. M_1 檢查每位成員的新會議金鑰 (K_{new}) 是否相同

假使沒有成員離開或加入會議的情況維持了一定的時間後，便可運用金鑰更新演算法：

(1).成員 M_1 會產生一個新的貢獻值 S_1'' 及一個新的隨機數值 $nonce_1$ ，並計算 $E_p(ID_1 \| S_1'' \| nonce_1)$ 後，利用廣播(broadcast)的方式將 ID_1 、 $E_p(ID_1 \| S_1'' \| nonce_1)$ 送給其它成員 M_i (i = 2,3,…,11)。

(2).成員 M_i (i = 2,3,…,11)收到成員 M_1 送來的資料後，便可以計算出新的會議金鑰 $K_{new} = K_{old} \oplus S_1''$ 。

(3). 成員 M_i 將 ID_i 、 $f(K_{new})$ 送給成員 M_1 。

(4).最後，成員 M_1 檢查其它成員所送來的會議金鑰是否一致，完成金鑰新演算法，結果如圖 3-4 所示。

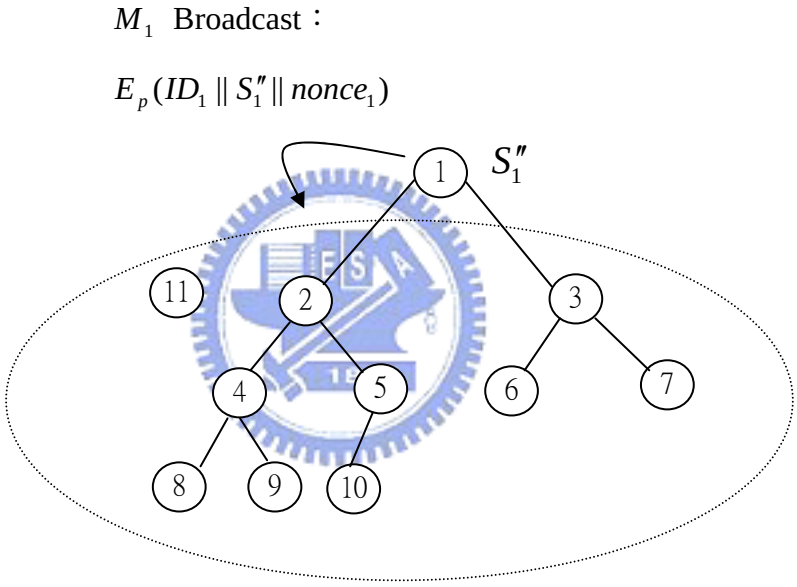


圖 3-4. 本論文所提的金鑰更新演算法執行結果

說明：

新的會議金鑰 $K_{new} = K_{old} \oplus S_1''$

四・分析與比較

在第三章中，本論文基於 Huang & Chang 的方法，提出了一個改良後的金鑰協同協定，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 安全性的分析比較

由於本論文所探討的金鑰協同協定是運用在動態會議的環境上，即成員可隨時動態地加入或離開會議，而綜合所看過的文獻資料[8][15]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，其次再將本論文所提的方法與 Huang & Chang 的方法做比較，說明是否真正解決了之前所分析出的問題。

4.1.1 動態會議金鑰協定條件驗證

以下幾點是一個動態會議金鑰協定所應俱有的安全條件，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需包含群組金鑰的安全。

- 群組金鑰安全 (Group Key Secrecy)

  保證對於一個消極型攻擊者而言，其想求出會議成員所產生的群組金鑰，為計算上之不可能。

- 金鑰獨立性 (Key Independence)

  保證對於一個消極型攻擊者而言，若知道任何一部份的群組金鑰資訊，其仍無法由此推出群組金鑰的其它部份資訊。

- 前推安全性 (Forward Security)

  保證對於一個已知舊有群組金鑰的消極型攻擊者而言，其無法利用此資訊計算或推出未來的群組金鑰。

■ 後推安全性 (Backward Security)

保證對於一個已知現有群組金鑰的消極型攻擊者而言，其無法利用此資訊計算或推出過去的群組金鑰。

在動態會議金鑰協定條件驗證中：

(1).關於群組金鑰安全方面，本論文所提的方法採用密碼 P 對成員的貢獻值及隨機產生的 *nonce* 值做加密，且唯有合法的成員才知道密碼 P ，對於一個非法的使用者而言，雖然可以取得合法成員所送之封包，但由於其不知道密碼 P ，加上受制於 *nonce* 的影響，仍無法解出其中所含成員的貢獻值，也就無法計算出最後的群組金鑰。

(2).關於金鑰獨立性方面，本論文所提的方法中群組金鑰是由所有成員的貢獻值經由 XOR 運算後所產生的結果，沒有一個成員可以僅靠部份成員的貢獻值便計算出會議金鑰。

(3).關於前推安全性與後推安全性方面，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 Huang 與 Chang 方法中的架構之上，且沒有變更其成員架構，因此可以採用 Huang 與 Chang 所提的成員加入及離開演算法[8]，來處理會環境中成員加入及離開的動作，以達到前推安全性與後推安全性。

4.1.2 與 Huang & Chang 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克服了該問題；同樣地，在一個 Ad Hoc 無線網路的環境下，配合本論文所提的方法做 Key Agreement 時，假設一個安全攻擊者有能力擷取到此協定過程的所有封包，則其可以擷取到的封包資料列於表 4-1 如下：

表 4-1. 執行本論文所提的方法時，攻擊者可收集到的所有封包

資料	索引值
$E_p(ID_i \| nonce_i)$	i = 2,3,…,n-1
$E_p(ID_{pt} \| nonce_c \| nonce_{pt})$	pt <= (n-1)/2 ; c > (n-1)/2
$E_p(ID_c \| nonce_{pt} \| K'_c)$	c > (n-1)/2
$E_p(ID_1 \| \pi \| nonce_1)$	
$E_{f(p\|\|\pi)}(S_n \| nonce_n)$	
$f(K_i)$	i = 1,2,…,n-1
ID_i	i = 1,2,…,n
註：n = 成員數目 ； 若 c = i 則 pt = $\lfloor i/2 \rfloor$	

以下驗證本論文所改良後的結果是可以解決 3.1 節所分析出的問題：

✓ 避免重送攻擊 (Replay Attack)

在本論文所提的方法中，多了用 *nonce* 及三方交握概念所達成的相互身份認證(mutual authentication)的機制，且 *nonce* 值是每次送出資料之前所產生的隨機數，每次傳送時皆會不同，當攻擊者利用其所竊取的封包做重送攻擊時，被攻擊的目標成員 M_i 雖然可用密碼 P 將密文正確反解，但此時可能會發現有二種情形：1. *nonce* 值和上次來自同一成員的 *nonce* 值重覆。或 2. 來源 *ID* 與密文中的 *ID* 不同。因此 M_i 並不會誤認攻擊者為合法成員而上當，致使攻擊者的重送攻擊失敗。

舉例說明：

如圖 4-1 所示，可以觀察出，假若有一個非法的惡意攻擊者 A 擷取到成員 4 送給成員 2 的所有封包資料 ID_4 、 $E_p(ID_4 \| nonce_4)$ 、 $E_p(ID_4 \| nonce_2 \| K_4')$，此攻擊者若謊稱自己是成員 4，而將此封包資料重送給成員 2，則成員 2 解密後得到 $nonce_4$ 且發現 $nonce_4$ 重覆，便判斷資料來源不合法；另一種情形，若攻擊者謊稱自己是合法的成員 2，想對目成員 1 做重送攻擊，由於其沒有合法的密碼 P ，只能將 ID_2 、 $E_p(ID_4 \| nonce_4)$ 送給成員 1，當成員 1 解開密文後發現 $ID_4 \neq ID_2$ ，也可判斷出資料來源不合法，因此運用本論文所提的方法，可避免重送攻擊的問題。

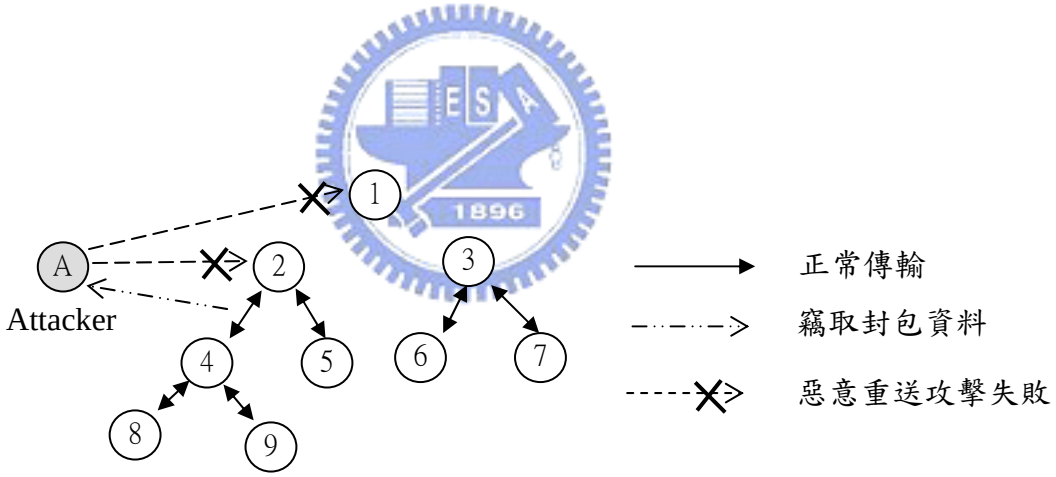


圖 4-1. 重送攻擊失敗示意圖

✓ 避免密碼猜測攻擊 (Password Guessing Attack)

在本論文所提的方法中，攻擊者所能竊取到的封包資料除了 ID_i 為明文、 $f(K_i)$ 為雜湊函式處理過後的值之外，其餘皆為加密過後的密文，而構成密文的元素中，由於對稱式加密用的金鑰：密碼 P ，

及隨機值*nonce*皆為攻擊者所不知，因此攻擊者無法利用明文攻擊或密文攻擊來求得密碼P。

- ✓ 改善執行效率上的問題

  關於本論文在 3-1 節所述的執行效率上的問題，將在下一小節做詳細的分析。

4.2 效率的分析比較

由於本論文所提的協定之運用環境是 Ad Hoc 無線網路，在此環境之下除了安全面的議題之外，如何讓無線終端設備的使用更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，包括完成協定的傳輸效率及運算複雜度，在合於安全性的要求之內，最理想的是整個協定的傳輸效率愈快愈好，而運算複雜度愈低愈好，綜合所參考的文獻[8][9]，以下有幾個指標，常被用來衡量金鑰協定的效率：

[傳輸效率衡量]

- ✓ 回合數 (Round)

  回合數是指在會議金鑰建立的過程中，所有參與會議的群組成員為了完成整個金鑰協定而必須執行的回合次數，本論文在此是以同步回合次數(Synchronous Rounds)為衡量的指標，即假設不同成員間的合作可以同步進行。

- ✓ 訊息量 (Message)

  當群組中的一位成員將一份資料封包傳給另外一個群組成員時，計為訊息量的一個基本單位，此訊息量指標的值即是指當金鑰協定執行的過程中，所有會議成員依據金鑰協定所需傳遞訊息次數的總合，本論文在此又將其細分成群體播送(multicast)及單點播送(unicast)二種。

- ✓ 合併訊息量是否增加 (Combined Message Size Grows)

  當成員間合作時，某成員送給下一成員的資料往往是利用前一驟時與前

一成員合作後的結果所計算產生，此指標便是判斷合作過程中，成員間傳遞的資料量是否隨著合作次數的增加而增加。

[運算複雜度衡量]

- ✓ Diffie-Hellman 金鑰交換(DH Key Exchange)次數

  在所探討比較的會議金鑰協定中，大部份都是基於 Diffie-Hellman 金鑰交換的方法，而此衡量指標便是記錄整個金鑰協定的執過程中，執行到 Diffie-Hellman 金鑰交換的次數。

- ✓ 運算處理 (Computation)

  指參與會議的各別成員，為了完成此金鑰協定所需執行的運算，由於不同的金鑰協定所利用到的運算方法可能會不同，而這個指標可以看出一個成員在運算上的負擔，愈簡單的運算對於設備省電愈有幫助。

通常在一個會議金鑰協定中，金鑰建立階段是最費時費力的，因為所有會議的成員都必須參與該協定，所以依據前述的效率衡量指標，本論文對於金鑰協定中產生起始金鑰的演算法，利用一個表格列出本論文所探討過的金鑰協定與本論文所提出的金鑰協定之比較：

表 4-2. 位於表 4-3.中的符號說明

符號	說明
n	群組成員的數量
i	成員的代號
H	雜湊函式(hash function)的執行
X	XOR 運算的執行
E	對稱式加密法中加密動作的執行
D	對稱式加密法中解密動作的執行
E_x	指數運算

表 4-3. 會議金鑰協定之效率評估比較

演算法 / 衡量指標	G-DH.2	Hypercube	DH-LKH	Huang & Chang	**The proposed method**
Rounds	n	$\log_2 n$	$\log_2 n$	$\log_2 n + 4$	$3\log_2 n + 3$
Multicast Messages	1	0	$\log_2 n$	2	2
Unicast Messages	n-1	$n \log_2 n$	0	3n-4	4n-7
Combined Message Size Grows	Y	N	Y	Y	N
DH Key Exchange	n	$(\log_2 n)/2$	$\log_2 n - 1$	0	0
Computation	IF i < n ： $(i+1)E_x$ IF i = n ： nE_x	IF i <= n ： $(\log_2 n)E_x$	IF i <= n： $(\log_2 n + 1)E_x$	IF i < n： 3H+1E+1D+4X IF i = n： 1H+(n-1)E+2X	IF i < n： 2H+4E+4D+3X IF i = n： 1H+1E+1D+1X

經由上述表 4-3 的效率評估比較，做了以下的分析：

1. 在回合數方面：

本論文所提方法的合作回合數比 GDH.2 來得少，但會比 Hypercube、DH-LKH、Huang & Chang 的方法來得多，這是因為本論文所提的方法中多加入了成員間的相互身份認證(Mutual Authentication)，所以必需多耗些回

合，但卻可以達到避免某些惡意攻擊之效用。

2. 在傳輸訊息量方面：

在群播的訊息量方面，雖然本論文所提的方法比 GDH.2 多一次，與 Huang & Chang 方法同樣多次，但就單一個廣播訊息的內容大小而言，其兩者的方法皆會受群組成員數量的影響，而本論文所提的方法並不會，即同樣一次的廣播，所傳送的訊息內容大小是小很多的；在單點播送的訊息量方面，本論文所提方法之傳送的資料量會比其它的方法大，這也是因為多加入成員相互身份認證機制的原故。

3. 在合併訊息量是否增加方面：

在 GDH.2、DH-LKH、Huang & Chang 的方法中，某些成員所傳送的資料量，是將部份資訊合併後一起送出，因此隨著回合的進行，所傳送的資料量會愈來愈大，但本論文所提的方法中，每一位成員所傳送的資料量皆是固定的，除了可提升成員合作時的傳輸效率之外，也較不容易有傳送資料遺失的問題。

4. 在 DH 金鑰交換次數方面：

由於 GDH.2、Hypercube、DH-LKH 方法是基於 Diffie-Hellman 金鑰交換的基礎之上，所以皆會用到一次以上的 DH 金鑰交換運算，即運算複雜度較高的模指數運算，而本論文所提的方法與 Huang & Chang 的方法同樣是以 XOR 運算為基礎，完全不會用到 DH 金鑰交換運算，就運算速度上而言是較有效率的。

5. 在運算處理方面：

在 GDH.2、Hypercube、DH-LKH 方法中，每個成員所需的運算次數皆會隨著成員數目的增加而遞增，在 Huang & Chang 的方法中，其最後一個秘密貢獻者(成員 M_n)，也是會有這樣的問題，而本論文所提的方法中，每一個成員的運算次數皆是常數，就算是最後一個秘密貢獻者其計算次數也與成員數目無關，並不會隨著成員數目的增加而遞增，因此當成員總數目很大時，運算效率上的改善相當顯著。

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 Ad Hoc 的網路環境上較適用金鑰協同協定，而不採用有 CA 的架構的金鑰協定，也了解利用 XOR 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 Huang & Chang 的方法架構之上，提出一個改良後的金鑰協同協定，除了修改其金鑰起始階段的演算法之外，還增加了金鑰定時更新機制以加強安全度，並且做了安全面及效率面的分析與比較，在安全面方面，證明本論文所提的方法除了符合會議金鑰協定的安全條件之外，也解決了 Huang & Chang 方法所遭受重送攻擊與密碼猜測攻擊，大大增加了安全上的強度；在效率面方面，本論文所提的方法方法保持 Huang & Chang 方法的精神，同樣利用 XOR 運算取代 Diffie-Hellman 的模指數運算來產生會議金鑰，可以增加運算上的效率，但不同的是，本論文所提的方法方法中，單一傳輸訊息量並不會受成員數目的影響，而是單一固定量，除了可增加合作傳輸上的效率之外，還可減少傳輸訊息遺失、錯誤的問題。

5.2 未來研究方向

本論文所提金鑰協定的應用環境是一個密閉空間，如：會議室・・等，並假設此空間內的成員皆合法，主要防範的攻擊者為位於會議空間外的非法者，但若是想要增加金鑰協定的強健性(robust)，思考如何防範內部成員的攻擊也是一個值得研究的議題。

參考文獻

1. D. Wallner , E. Harder and R. Agee, “Key Management for Multicast: Issues and Architecture”, IETF Draft, July 1997.
2. D. Wallner , E. Harder and R. Agee, “Key Management for Multicast: Issues and Architecture”, RFC 2627 , 1999.
3. G. Yao , K. Ren , F. Bao , R. Deng and D. Feng. “Making the Key Agreement Protocol in Mobile Ad Hoc Network More Efficient”, In ACNS 2003, pp. 343-356. 2003.
4. K. Becker and U. Wille. “Communication Complexity of Group Key Distribution”, In ACM 5th Conference on Computer & Communications Security, pp.1-6, Nov, 1998.
5. L. Zhou and Z. Haas. “Securing Ad Hoc Networks”, IEEE Network , pp.24-30, Dec, 1999.
6. M. Steiner , G. Tsudik and M. Waidner. “Diffie-Hellman Key Distribution Extended to Group communication”, In ACM CCS’96, pp.31-37, 1996.
7. O. Rodeh , K. Birman and D. Dolev. “Optimized group rekey for group communication system.”, In Network and Distributed System Security , 2000.
8. R. J. Huang and R. C. Chang. “Key Agreement in Ad Hoc Networks”,In ISPA 2003, pp.382-390, 2003.
9. 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.
10. S. M. Bellovin and M. Merrit. “Encrypted key exchange: password-based protocols secure against dictionary attacks”, IEEE Symposium on Research in

Security and Privacy, pp.72-84, 1992.

11. S. M. Ghanem and H. A. Wahab. “A Simple XOR-based Technique for Distributing Group Key in Secure Multicasting”, IEEE Symposium on Computers and Communications, pp.166-171, 2000.
12. W. Dondeti , S. Mukherjee and A. Samal. “A distributed group key management scheme for secure many-to-many communication”, Tech. Rep. PINTL-TR-207-99, Department of Computer Science, University of Maryland.
13. W. Diffie and M. E. Hellman. “New Directions in Cryptography”, IEEE Transaction on Information Theory, Vol.IT-22, No.6, pp.644-654, Nov. 1976.
14. X. Li , Y. Wang and O. Frieder. “Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Network”, IEEE International Conference on Computer Communications and Networks, ICCCN 2002, pp.404-409, 2002.
15. Y. Kim , A. Perrig and G. Tsudik. “Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups”, In ACM CCS’00, pp.235-244, 2000.
16. 林峻立. “使用者通行碼之身份驗證與金鑰交換協定”, Communications of the CCISA, Vol.9, No3, June 2003, pp.43-52.