國立交通大學

理學院科技與數位學習學程

碩士論文

安全即時通訊系統之設計與實作

The Design and Implementation of A Secure Instant Messaging Service

研究生：林德政

指導教授：李榮耀　教授

中華民國　一百　年六月

安全即時通訊系統之設計與實作
The Design and Implementation of A Secure Instant Messaging Service

研 究 生：林德政　　Student：De-Zheng Lin

指導教授：李榮耀　　Advisor：Jong-Eao Lee

國 立 交 通 大 學

理學院科技與數位學習學程

碩 士 論 文

A Thesis

Submitted to Degree Program of E-Learning

College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Degree Program of E-Learning

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年六月

安全即時通訊系統之設計與實作

學生：林德政　　指導教授：李榮耀 教授

國立交通大學理學院科技與數位學習學程

摘要

即時通訊(Instant Messaging，IM)提供使用者在網路上交換即時文字訊息、語音、視訊、檔案等通訊服務，對許多網路使用者而言，已經成為日常生活不可或缺的聯絡工具。目前多數的即時通訊軟體多以明文的方式傳送即時訊息，不僅如此，在伺服器與使用者之間亦缺乏安全完善的驗證機制。而這些漏洞容易導致即時通訊系統容易遭受竊聽、使用者假冒與伺服器偽裝等網路攻擊，對使用者的隱私造成威脅。有鑑於此，如何提供完善的驗證機制，並對通訊的內容提供加密保護的安全機制，以提升即時通訊系統的安全性，已是刻不容緩。

本論文以學者 Lee 等人所提出的三方驗證金鑰交換協定為研究基礎，設計一套安全的即時通訊協定，並在 Java 平台實作本即時通訊系統，以驗證本研究所提出的即時通訊協定是可行的。在我們的即時通訊協定中，不僅讓伺服器與使用者之間能相互驗證彼此的身份，並在驗證完雙方身份無誤之後，讓通訊的兩位使用者間可以建立一把階段性共通金鑰(Session Key)，而這把金鑰可以用來對即時通訊的內容加密，防止即時訊息透過網路傳送時，遭到攻擊者擷取、利用，以符合驗證性、機密性、完整性、前推私密性等安全需求，且在安全分析中，我們分析了幾種目前常見的網路攻擊方法，以提供使用者更安全的即時通訊協定。

關鍵字：即時通訊、密碼學、驗證、金鑰交換

The Design and Implementation of a Secure Instant Messaging Service

Student：De-Zheng Lin Advisors：Dr. Jong-Eao Lee

Degree Program of E-Learning
College of Science
National Chiao Tung University

ABSTRACT

IM (Instant Messaging) offers users the service of exchanging text messages, audios, videos and files instantly on a network. For many network users, IM becomes an indispensable tool for daily communications. So far, most sets of the IM software transmit instant messages in plaintext, and without a secure flawless authentication scheme between the server and users. These Flaws make the IM systems prone to eavesdrop attack, impersonation attack and server spoofing attack, which poses a threat to the users' privacy. So it is necessary to enhance the security of IM systems by providing a flawless authentication scheme and the encryption approaches to secure the IM content in no time.

This study, based on the three-party encrypted key exchange (EKE) protocol of Lee et al., designs a secure IM protocol and implements the IM system in the Java platform to verify that the proposed IM protocol is feasible. The proposed IM protocol not only enables the server and users to authenticate each other mutually but also generates a session key between the two users on communication after authenticating the validity of their identification. This session key is used to encrypt the IM content, protecting the instant messages sent on a network from being intercepted or misused by attackers, and therefore, meets the security requirements: authentication, confidentiality, integrity, and forward secrecy. In the security analysis, we also analyze several commom network attack methods and provide users with a more secure IM protocol.

Keyword：Instant Messaging, Cryptography, Authentication, Key Exchange

誌　謝

這本論文能夠完成，要感謝許多人的幫忙。首先要感謝李榮耀教授的指導，在三年碩士進修的期間，老師的諄諄教誨，不論是研究所的課業學習，或是論文研究方面，都對我有很大的助益。其次要感謝的是口試委員，感謝委員們花費時間與心力來審查這本論文，並在百忙之中抽空前來參加我的論文口試，給予我寶貴的指導與建議，讓這本論文的內容更加完善。此外，還要感謝蔡佳倫學長在專業領域上的指導與協助，學長對研究的熱衷執著，以及在學術專業上的成就，深深地影響了我，在與學長的討論中，我深刻地體悟到學術研究的態度與方法，不論在期刊的閱讀，或是論文撰寫的技巧，都令我獲益良多，學長的鼓勵是督促我成長進步的動力，在學長的熱心指導下，我才能完成這一本論文。

接著要感謝專班的同學瑜顯、文羿、秀霞、文熾的鼓勵，還有任教學校的同事敏晧、義淵、士譽、宏遠、源盛、雅夢、懿君，在學校的教學與行政業務上給予我許多幫助，最後要感謝我的父母及家人，在我利用週末上課，無暇回家的時候，總是默默的支持我。在此要對陪伴我走過這三年研究所進修生涯的家人、指導教授、學長、專班同學與任教學校的同事，致上崇高的感激與敬意，因為有你們的鼓勵與協助，我才能完成這本論文，真心的感謝你們。

目　錄

表目錄

圖目錄

E S A
1896

第一章 緒論

即時通訊軟體提供使用者能夠在網路上互相交換即時訊息的通訊服務，已逐漸成為人們日常生活的網路應用工具。本研究旨在開發一個安全的即時通訊系統，利用各種不同的密碼學的技術，使進行通訊的雙方，可以透過一個可信賴的第三方互相驗證對方的身份，並且在確認對方身分之後，產生一把階段性共通金鑰，以保護即時通訊訊息的安全性。本章針對研究背景、研究動機、研究目的與論文架構做探討。

1.1 研究背景與動機

隨著電腦硬體與網際網路的發展普及，有越來越多的應用系統建構在網路上以提供更方便的服務平台，例如網路報稅繳稅、電子商務、遠距學習等。這些系統讓遠端的使用者可以藉由網路傳送資訊，使用許多線上的服務功能，縮短空間範圍的距離，不受時間地點的限制，以提升生活的品質。

即時通訊(Instant Messaging，IM)提供使用者在網路上交換即時文字訊息、語音、視訊、檔案傳輸等通訊服務。目前已有 Windows Live Messenger、Yahoo!Messenger、Skype、ICQ、Google Talk 等即時通訊系統提供網友註冊使用。它已經漸漸地進入到我們的日常生活之中，對許多的網路使用者而言，即時通訊軟體改變了人們的生活方式，已經成為日常生活不可或缺的溝通聯絡工具，更有研究指出，一般民眾在使用網路時，除了瀏覽網頁與收發電子郵件外，最常使用的便是即時通訊服務，由此可知它的普及性[19]。

然而，隨著即時通訊軟體的普遍使用，隨之而來的安全問題也逐漸浮現，近幾年來隨著使用的人數變多，網路駭客也漸漸的開始攻擊起即時通訊軟體，近年來比較著名的攻擊事件有：2010 年 3 月間，台灣的 MSN 即時通訊系統發生大當機，許多人的密碼遭破解，帳號被盜用，重要的個人資料遭到外洩[10]；同年 5 月，又發生使用者被假冒的聯絡人帳號詐騙金錢財物，損失甚鉅[24]。此外，目前多數的即時通訊軟體在使用者登入系統後，多以明文的方式傳送即時通訊訊息，對通訊的內容沒有任何加密保護的機制。攻擊者只要監聽網路上的封包，即可在網路傳輸的不安全環境下，得知使用者間通訊的內容，對使用者的隱私造成威脅。有鑑於此，如何提供即時通訊軟體可以驗證即時通訊伺服器與使用者的身份，並對即時通訊的內容提供加密保護，以提升系統的安全性，已是刻不容緩。

本研究以理論研究為基礎，提出一個安全的即時通訊協定，並將研究結果以實作的方式來檢視本研究所提出的方法是可行的，期許透過本研究，可以讓即時通訊系統更加的安全、方便。

1.2 研究目的

本研究的目的是藉由網路技術與密碼學的加密演算法，在 Java 平台開發安全的即時通訊系統，提供使用者安全的即時通訊環境。主要目的如下：

1. 利用相關的密碼學演算法，設計安全的即時通訊協定。
2. 在 Java 平台採用 Diffie-Hellman 公開金鑰交換演算法、RSA 公開金鑰密碼系統、Triple-DES 三重資料加密標準與 SHA-1 單向雜湊函數等密碼學技術，開發安全的即時通訊系統。

1.3 論文架構

本論文共分為六個章節，各章節的內容說明如下：

第一章為緒論，說明本篇論文的研究背景、研究動機與研究目的。

第二章介紹相關知識與過去重要的文獻回顧。

第三章介紹本研究設計的即時通訊協定。

第四章針對本研究設計的即時通訊協定做安全性分析與效能分析。

第五章說明即時通訊系統實作，包含伺服器與使用者的開發環境及系統測試。

第六章提出本研究的結論與建議。

第二章 文獻探討

本章就即時通訊系統相關的理論與技術加以探討，包括即時通訊服務、身份驗證、金鑰交換協定、常見網路攻擊方法、相關密碼學理論技術與相關研究等。

2.1 即時通訊服務

網際網路工程任務小組(The Internet Engineering Task Force，IETF)在 2000 年成立了即時通訊與線上狀態協定工作小組(Instant Messaging and Presence Protocol Working Group，IMPP WG)，提出了 RFC 2778[2]標準，定義了即時通訊系統的兩種服務模式，分別為線上狀態服務(Presence Service)與即時通訊服務(Instant Messaging Service)。線上狀態服務可以讓使用者知道其他聯絡人目前在即時通訊系統的使用狀態，如圖 2-1 所示。

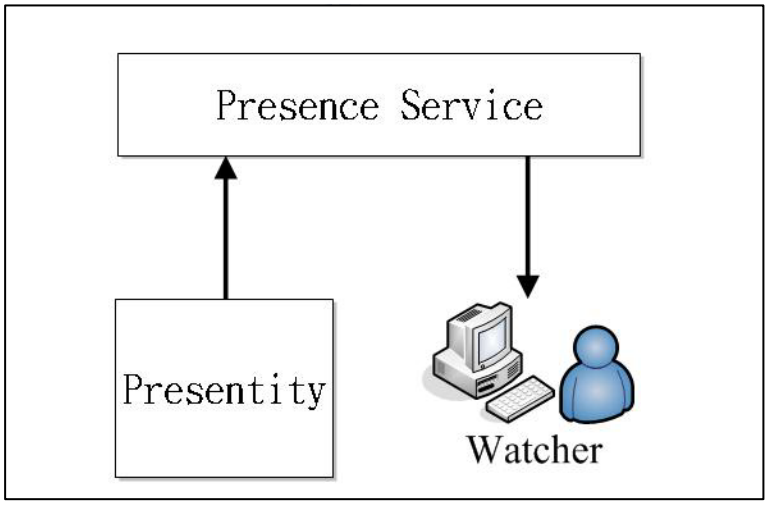

圖 2-1：RFC 2778 標準定義即時通訊系統線上狀態服務
資料來源：[2]

即時通訊服務則提供使用者間即時訊息的傳送，當一個傳送者要傳送即時訊息給接收者時，即時通訊服務先將即時訊息儲存在接收者的 Instant Box，接收者再使用即時通訊軟體接收在 Instant Box 上的即時訊息，如圖 2-2 所示。

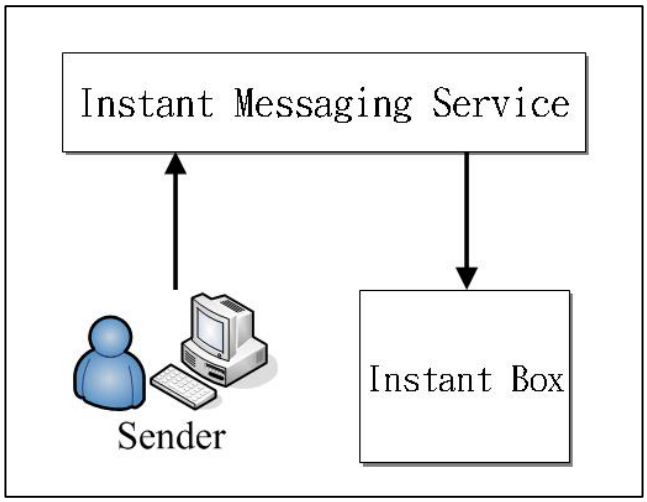

圖 2-2：RFC 2778 標準定義即時通訊系統即時通訊服務
資料來源：[2]

即時通訊的方便性在於，使用者可以透過即時通訊伺服器取得聯絡人的名單(Contact List)與線上狀態(Presence)，另外，使用者之間還可以即時傳遞文字訊息、音訊、視訊與檔案，如圖 2-3 所示。所以，即時通訊系統的架構基本上是由一部伺服器與兩位以上的使用者所組成，除了使用者登入伺服器的身份驗證外，還必須考量到使用者與使用者之間傳送即時訊息的通訊安全。因此，一個安全的即時通訊系統必須提供伺服器與使用者的相互驗證，並確保所有使用者相互之間即時訊息的通訊安全，以滿足驗證性、機密性、完整性與前推私密性[17] [21]等安全需求。

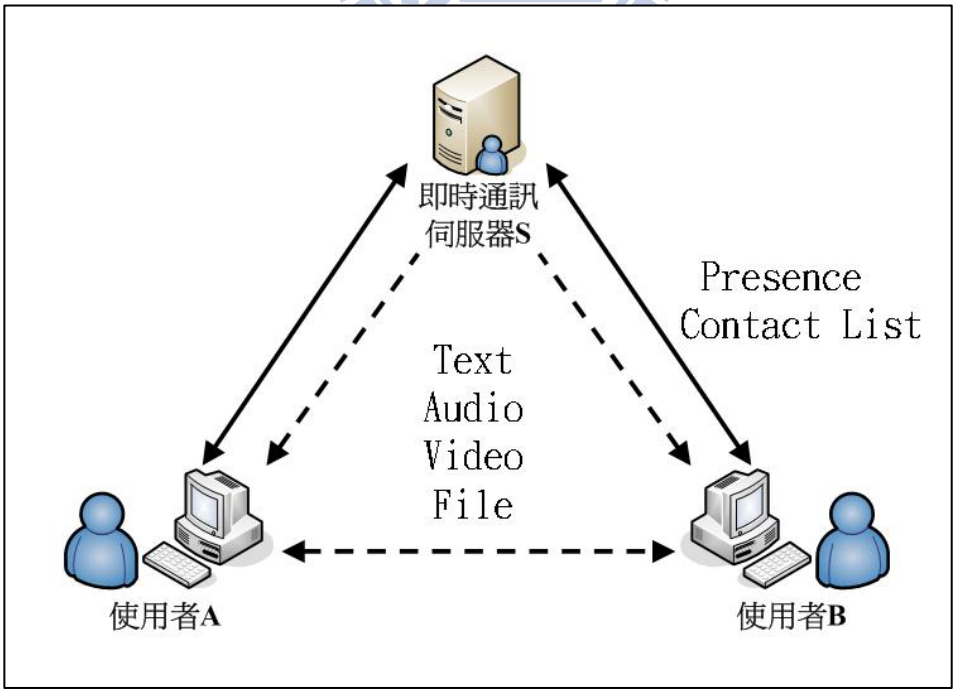

圖 2-3：即時通訊服務的三方通訊模型
資料來源：[21]

目前的即時通訊協定有兩個主要標準：SIMPLE(SIP for Instant Messaging and Presence Leveraging Extensions)與 XMPP(Extensible Messaging and Presence Protocol)。SIMPLE 以 SIP(Session Initiation Protocol)為基礎，以點對點的方式傳送即時訊息，被廣泛的應用在多媒體通訊，如網路電話、視訊等。XMPP 以 XML(eXtensible Markup Language)為通訊資料格式，依照主從式架構傳送即時訊息[21]。

2.2 身份驗證與金鑰交換協定

隨著硬體技術的進步與網際網路的普及，許多資源可以藉由網路的管道共同分享。網路帶來資料傳輸的便利性，提供人們許多無遠弗屆的服務，卻也帶來許多安全上的問題。在開放的網路環境中，資料的傳輸如果沒有適當的保護機制，容易被惡意的攻擊者竊聽、利用，進一步對使用者的隱私與權益造成嚴重的威脅。因此，如何提供身份驗證(Authentication)與交換加解密金鑰 (Encrypted/Decrypted Key Exchange)一直是網路安全中很重要的非常重要的研究議題[18]。

2.2.1 驗證(Authentication)

在實際的日常生活中，我們可以辨識一個人的五官長相、指紋，或是根據證件、識別物等，來驗證他的身份；但在網路環境上，因為無法見面，所以無法根據這些對方的特徵或持有物來驗證對方的身份。所以，要在網路上驗證對方的身分是十分困難的，因此，如何提供伺服器去驗證使用者的身分以及讓使用者去驗證伺服器的身分，便成了一個很重要的議題，而這樣的議題我們亦稱為驗證協定(Authentication Protocol)[27]。

在網路上以密碼(Password)來驗證遠端使用者身份的方法，已經是相當普遍的機制。早期用來驗證身份的密碼為一個固定的值，當使用者傳送他的驗證資料給伺服器時，如果沒有適當的加密保護機制，容易被惡意的攻擊者竊聽、利用，然後用來假冒合法使用者登入遠端的伺服器，從事非法的行為，對使用者與伺服器造成危害。為了防止身份驗證的密碼在網路傳輸時遭到竊聽，必須藉由密碼學的技術與驗證協定的改良，動態地改變驗證資料，使密碼在傳輸的過程中成為一個不固定的值。然而，在資訊技術與硬體運算速度不斷地進步下，以密碼來驗證遠端使用者身份的機制，還是存在許多的風險與漏洞，安全性的提升依舊是刻不容緩[18]。

目前在網路上驗證遠端使用者身份的模式，大致歸類為下面三種型態[27][25]：

1. 單向驗證(One-Way Authentication)

使用者先在伺服器註冊帳號密碼，當使用者要登入伺服器時，必須提供帳號與密碼給伺服器，由伺服器驗證使用者的身份，是最簡單而且常見的驗證方式，如圖 2-4 所示。然而，這樣的驗證方式存在伺服器偽裝攻擊的安全性問題，因為使用者在這樣的型態下，並無法驗證伺服器的身分，所以攻擊者可以在網路上偽裝成一部合法的伺服器，去欺騙使用者。

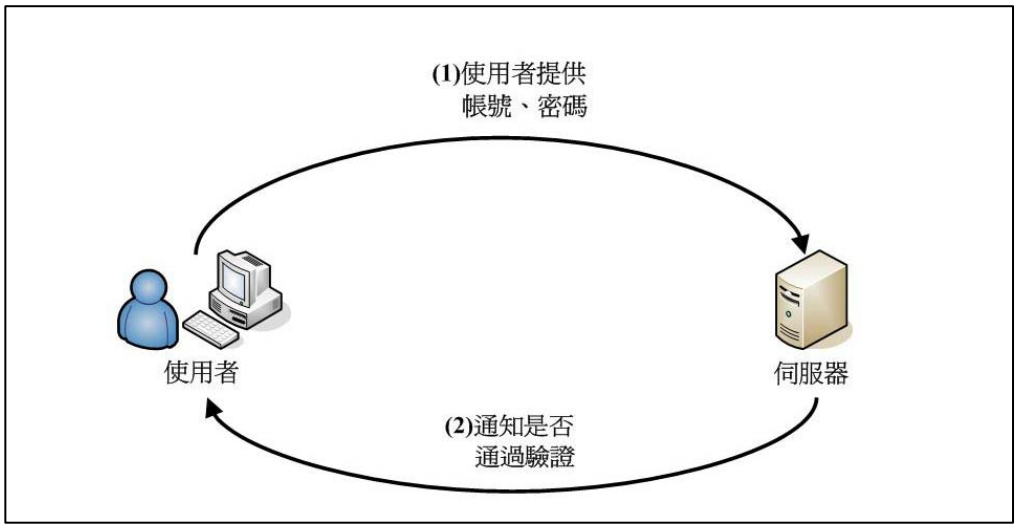

圖 2-4：單向驗證
資料來源：[25]

2.雙向驗證(Mutual Authentication)

雙向驗證主要是用來改善單向驗證的缺點，當使用者提供帳號與密碼給伺服器，伺服器亦提供帳號、密碼給使用者，讓雙方可以相互驗證對方的身份，如圖 2-5 所示。不過，在雙向驗證的機制中，使用者與伺服器必須妥善維護對方的驗證資料，若使用者的驗證資料遭到攻擊者竊聽，攻擊者便可假冒使用者的身份登入伺服器；若伺服器的驗證資料不慎洩漏，攻擊者便可利用這一份資料，偽裝成一部合法的伺服器，欺騙使用者。

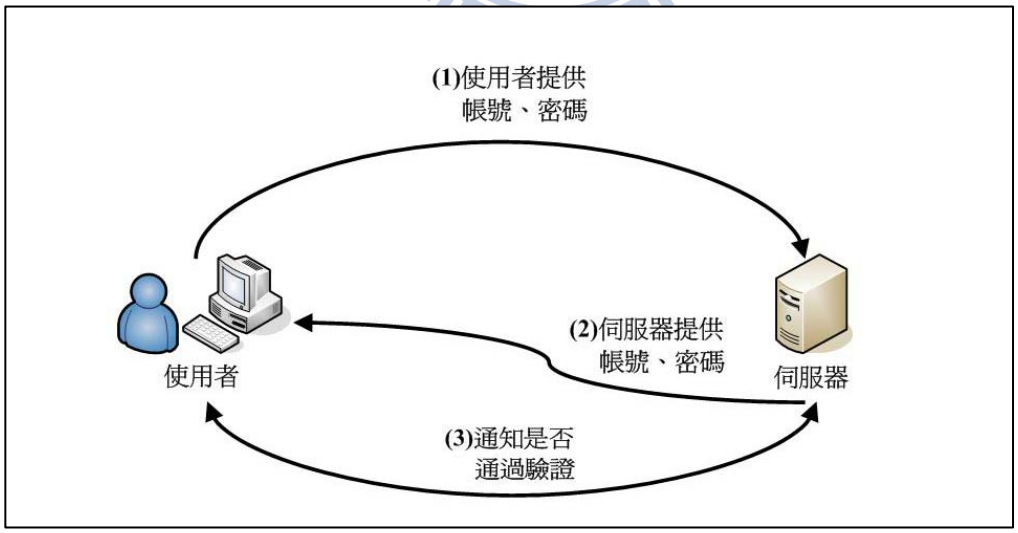

圖 2-5：雙向驗證
資料來源：[25]

3.經由公正、可信賴的第三方驗證(Trusted Third-Party Authentication)

使用者與伺服器各自提供帳號與密碼給一個公正、可信賴的第三方，可信賴的第三方在確認使用者與伺服器的身份後，會提供雙方一份驗證資料，使用者與伺服器可以利用第三方提供的驗證資料，來驗證對方的身份，如圖 2-6 所示。在這樣的驗證機制下，第三方的安全性就變得相當重要，若公正的第三方遭到攻擊入侵，那麼使用者與伺服器之間的身份驗證將不再安全。

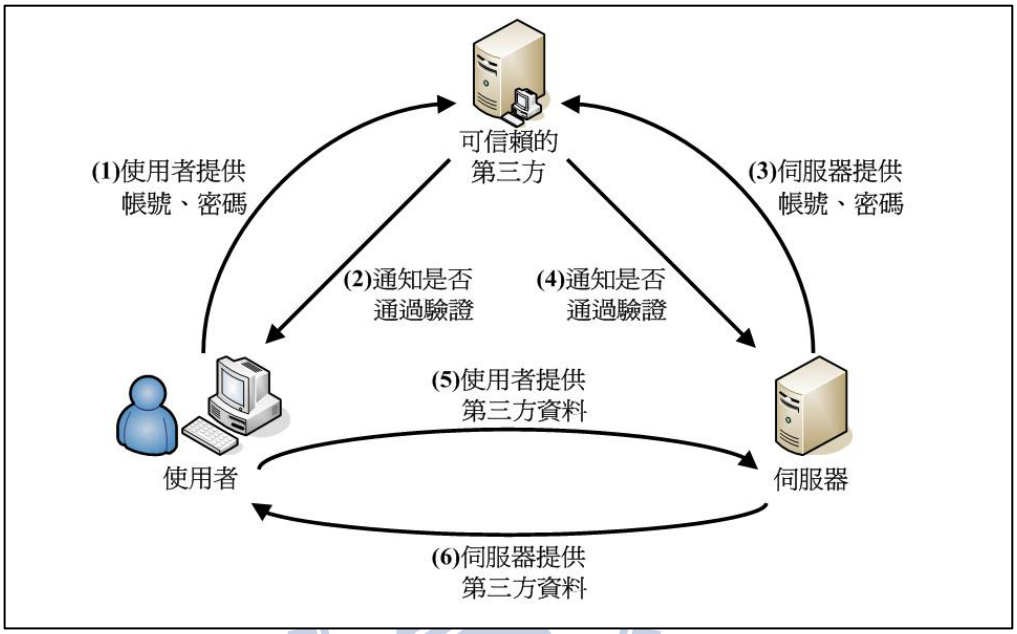

圖 2-6：經由公正可信賴的第三方驗證

資料來源：[25]

2.2.2 金鑰交換協定(Key Exchange Protocol)

經由網路通訊的訊息明文，必須要經過金鑰的加密保護，才能確保資料在傳送的過程不被攻擊者竊聽得知通訊內容。在不安全的網路環境中，驗證通訊雙方的身份，並且透過密碼學技術與安全的機制，讓參與通訊的成員能夠協調一把共通的加解密金鑰(Session Key)，這個機制即為金鑰交換協定(Key Exchange Protocol)，是網路安全一項很重要的議題。

一個好的金鑰交換協定，必須具備下列的安全需求[22]：

1. 驗證性(Authentication)

在網路上通訊的雙方，必須確認對方的身份。接收者要能透過接收到的訊息，來確定對方使用者的身分。

2. 機密性(Confidentiality)

資料透過網路傳送時，必須確保傳送的資料不會洩漏。在網路上傳送的訊息必須經過加密處理，並且只有正確的接收者才能知道訊息的內容。攻擊者即使可以竊聽網路傳

送的封包，也無法得知訊息的內容。

3. 完整性(Integrity)

傳送者傳送的訊息，與接收者所收到的訊息必須要完全相同，當攻擊者企圖對網路傳輸的訊息加以修改，或是偽造假訊息傳送給接收者時，接收者必須能夠確認接收到的訊息，是否經過竄改或偽造。

4. 前推私密性(Forward Secrecy)

假設傳送者與接受者共通的加解密金鑰不慎洩漏，不會影響到雙方先前使用過的共通金鑰安全性。攻擊者即使可以取得某一次的共通金鑰，也無法利用這一把金鑰去破解、推算先前的共通金鑰，進一步解開先前加密的訊息[17]。

2.3 網路攻擊方法

一般常見的網路攻擊方法有下列幾種：

1. 竊聽攻擊(Eavesdrop Attack)

由於網路是一個開放的環境，所有在網路上傳送的訊息都是公開的，攻擊者藉由監聽網路上傳送的封包，即可取得使用者登入伺服器的驗證資料。對於以傳統的帳號、密碼來進行使用者身份驗證的機制，這種攻擊特別有效。為了避免竊聽攻擊，必須利用密碼學的技術，將驗證資料經過加密以後再傳送到網路上[12]。然而，對使用者而言，即使將帳號、密碼等驗證資料加密後再傳遞，攻擊者一樣可以藉由竊聽攻擊擷取到加密後的訊息，再將這些加密過的驗證訊息重新傳送給伺服器，假冒合法的使用者登入伺服器，對使用者及系統造成危害。要有效防止竊聽攻擊的威脅，必須在訊息傳送時，加入時間戳記(Timestamp)或是使用隨機亂數(Nonce)，動態地改變傳送的訊息[25]。

2. 重送攻擊(Replay Attack)

攻擊者藉由網路竊聽技術，在網路上擷取使用者與伺服器之間的通訊紀錄及資料封包，並對這些資料進行分析、破解，從中找出使用者身份驗證的訊息，例如：帳號、密碼等，並在合法的時間內，將使用者的驗證資料重新傳送給伺服器，假冒合法使用者的身份登入伺服器，入侵系統。

要在網路上阻止這種類型的攻擊，可以在驗證資料加入時間戳記(Timestamp)，或是使用隨機亂數(Nonce)，動態地改變驗證訊息，使得每一次傳送的驗證訊息都不相同。攻擊者即使可以擷取到使用者的驗證資料，並將這些訊息重新傳送給伺服器，也無法通過伺服器的驗證[15]。

3. 密碼猜測攻擊(Password Guessing Attack)

在網路上要求遠端使用者輸入帳號、密碼，來驗證使用者的身份，是目前很常見網路驗證機制。然而，要求使用者要記憶冗長且無意義的密碼並不容易，多數使用者會選擇較簡短熟悉、日常生活容易記憶的數字，或是有意義的英文單字、片語來作為身分驗證的密碼，例如生日、電話號碼等，這時攻擊者便可利用密碼猜測的方式進行攻擊[25]。

Ding 與 Horster[9]將密碼猜測攻擊(Password Guessing Attack)分為以下三種類型：

(1)可偵測的線上密碼猜測攻擊(Detectable On-Line Password Guessing Attack)

攻擊者直接在網路上猜測使用者的密碼，嘗試通過伺服器的驗證，一般來說，猜測錯誤密碼的攻擊者會被伺服器偵測發覺，並且無法通過驗證，一般來說，這樣的攻擊方法並無法避免，要阻止這種類型的攻擊，可以在伺服器設立一個計數器，記錄使用者以錯誤密碼登入的次數，在一個固定時間內，如果有攻擊者以猜測錯誤的密碼密集登入，企圖通過伺服器的驗證，在達到一定的安全次數後，伺服器將會停止攻擊者的密碼驗證要求，並告知原帳號的合法使用者[16]。

(2)無法偵測的線上密碼猜測攻擊(Undetectable On-Line Password Guessing Attack)

無法偵測的線上密碼猜測攻擊的攻擊模式跟可偵測的線上密碼猜測攻擊在攻擊的手法上差不多，都是攻擊者直接在網路上猜測使用者的密碼，企圖通過伺服器的身份驗證協定，不過，與前面不同的，會發生這樣的攻擊，大部分是因為驗證協定並沒有提供伺服器可以偵測的到攻擊者目前正在猜測密碼的能力，因此，伺服器在沒有驗證使用者的情況下，直接回傳可能可以給攻擊者猜測密碼的驗證訊息，就容易讓攻擊者去檢驗猜測的密碼是否正確。

(3)離線密碼猜測攻擊(Off-Line Password Guessing Attack)

攻擊者不直接在網路上猜測使用者的密碼，而是在網路上擷取密碼的相關訊息，利用這些訊息，以離線的方式進行密碼猜測攻擊，間接猜測出正確的密碼。由於攻擊者並未對伺服器提出密碼驗證的要求，所以伺服器無法察覺攻擊者進行離線密碼猜測的攻擊行為[9][25]，所以這類型的密碼猜測攻擊是十分危險的。

4. 使用者假冒攻擊(Impersonation Attack)

攻擊者利用竊聽網路得到的訊息，經過處理，假造出一組合法使用者的驗證資料，通過伺服器的身份驗證，並透過這一個假造的合法使用者入侵系統，進行非法的資料存取，或是惡意竄改使用者的重要資料，對伺服器及使用者的資訊安全，造成嚴重的威脅。對伺服器或是網路管理人員而言，要察覺攻擊者假冒合法使用者登入系統的行為，是非常困難的[23][13][14]。

5. 伺服器偽裝攻擊(Server Spoofing Attack)

攻擊者在網路上偽裝成一部合法的伺服器，使用者在無法驗證伺服器的身份是否合法，或是驗證機制本身有安全漏洞的情況下，將使用者個人的帳號、密碼等驗證資料傳送給偽裝的伺服器。偽裝伺服器在取得使用者的驗證資料後，便可利用這一組合法的驗證資料登入正確的伺服器，進而修改合法使用者的驗證資料，或是入侵伺服器內部，取得其他使用者的私密資料，對合法使用者與伺服器造成危害。[23]

要有效的避免伺服器偽裝攻擊，使用者必須能夠對伺服器的身份加以驗證。常見的驗證機制為伺服器與使用者間的雙向驗證，或是透過一個公正、可信賴的第三方，來驗證伺服器與使用者雙方的身份[25]。

2.4 相關密碼學理論與技術

2.4.1 密碼系統(Cryptosystem)

密碼系統(Cryptosystem)即是將原來形式的明文(Plaintext)，經過某種特殊形式的處理，例如進行不可被破解的數學函數或數學計算等等方式，轉換成另一種偽裝隱藏的形式，而這種隱藏的形式通常我們稱之為密文(Ciphertext)。而密文，通常只有經過授權許可的人，才能將其轉換回原本的明文。通常來說，將明文轉換成密文，這個過程我們稱之為加密(Encryption)；相反的，將密文轉換回原本的明文，這個過程我們稱之為解密(Decryption)。

假設傳送者手上有一份明文要傳送給接收者，這一份明文除了接收者以外，不能讓其他人知道明文的內容。為了防止在傳送的過程中，明文遭到惡意第三者的攔截窺視，在明文傳送前，傳送者必須選擇一個參數，利用這個參數將原本的明文轉換成密文，再將密文傳送給接收者，這個參數即為加密金鑰(Encryption Key)；同樣地，接收者也必須持有一個參數，用來將密文還原為原本的明文，這個參數即為解密金鑰(Decryption Key)[22]，如圖 2-7 所示。

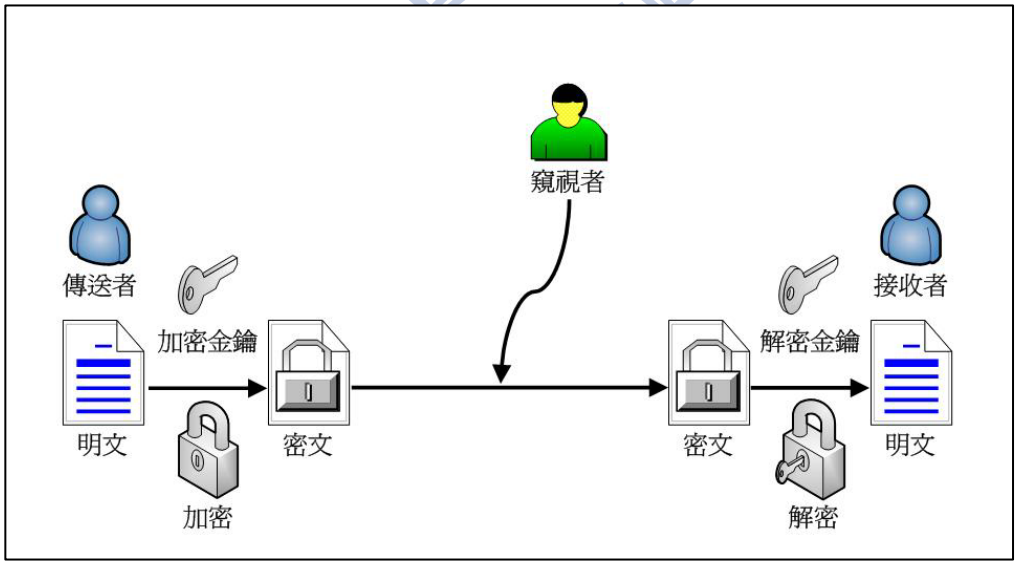

圖 2-7：密碼系統
資料來源：[22]

對傳送者與接收者而言，用來對密文解密的金鑰一旦洩漏，任何非法獲取解密金鑰的攻擊者，都可以輕易的破解加密過的資料，進而得知明文訊息的內容，接收者與傳送者之間的通訊將不再安全。因此，有人提出了非對稱性的金鑰加解密碼系統，因此，我們可以將密碼系統簡單的區分為對稱式密碼系統(Symmetric Cryptosystem)與非對稱式密碼系統(Asymmetric Cryptosystem)。

2.4.2 對稱式密碼系統(Symmetric Cryptosystem)

對稱式密碼系統(Symmetric Cryptosystem)又稱為私密金鑰密碼系統(Private-Key Cryptosystem)。在資料傳送前，傳送者與接收者必須事先協商一把共同的金鑰，這把金鑰用來加密及解密[15]。要進行通訊時，傳送者先用這把金鑰將明文加密成密文後，再傳送給接收者；接收者收到密文之後，用相同的金鑰對密文解密，就可以得到原先的明文。如圖 2-8 所示。

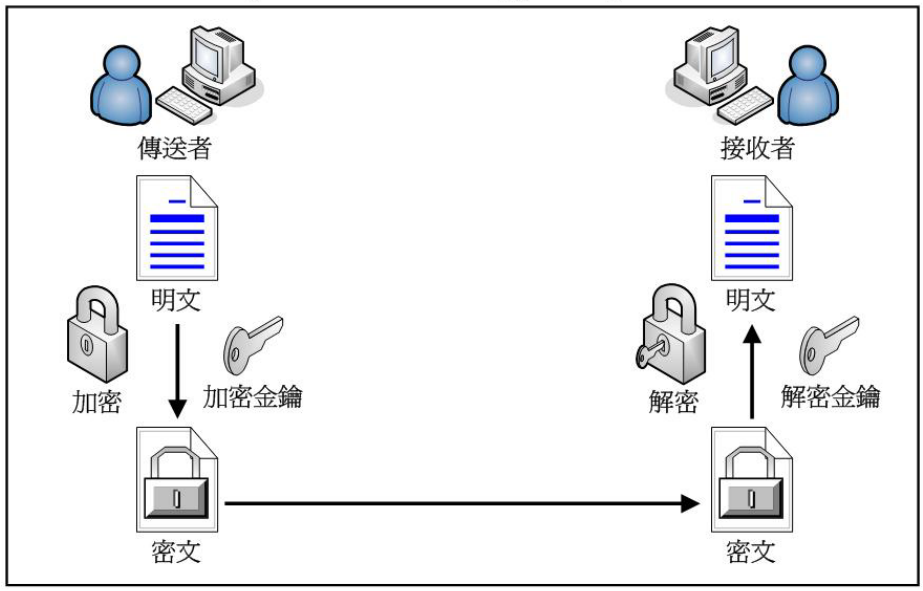

圖 2-8：對稱式密碼系統
資料來源：[11]

對稱式密碼系統的特色是用同一把金鑰來進行加密、解密，優點是加、解密的速度較快，適合用在大量資料的處理。使用對稱式密碼系統的傳送者與接收者，必須持有相同的加解密金鑰，如何協商這一把金鑰，是對稱式密碼系統的安全議題[6][26][11]。目前常見的對稱式加密演算法有 DES、Triple-DES、AES、IDEA、FEAL 等。

2.4.3 三重資料加密標準(Triple Data Encryption Standard)

1970 年初，美國 IBM 公司發展 LUCIFER 演算法，1977 年美國國家標準局採用 LUCIFER 演算法成為資料加密標準(Data Encryption Standard，DES)。DES 是一種區塊加密法，每次加密、解密的區塊大小為 64 位元。而在電腦運算速度發展越來越快的趨勢下，DES 的金鑰長度僅 56 位元，存在遭到暴力法破解的危機。為了避免金鑰遭到攻擊者的暴力破解，可以串接三個 DES 金鑰來使用，即所謂的三重資料加密標準(Triple-DES)。Triple-DES 的金鑰長度可以是 112 位元或 168 位元，較 DES 的金鑰長度僅 56 位元，安全性明顯提升許多[6][15][22]。

2.4.4 非對稱式密碼系統(Asymmetric Cryptosystem)

1976 年，美國兩位學者 Diffie 與 Hellmam[8]提出一個公開金鑰交換演算法，讓未曾見過面的兩個人，可以取得共通的金鑰，是公開金鑰最早的想法。公開金鑰密碼系統(Public-Key Cryptosystem)用來加密與解密的金鑰並不相同，因此又稱為非對稱式密碼系統(Asymmetric Cryptosystem)。

假設傳送者想要將資料安全的傳送給接收者，那麼接收者必須先將他的加密金鑰公開，傳送者取得接收者的公開加密金鑰後，使用這把金鑰將資料加密成密文後再傳送；接收者收到密文後，再使用自己的私密解密金鑰對密文解密，就能得到原先的資料。如圖 2-9 所示。

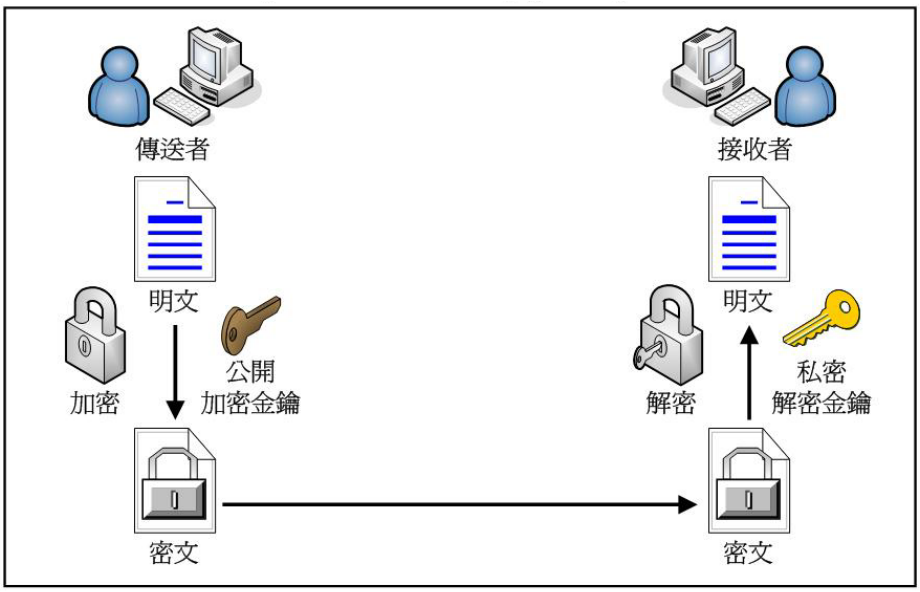

圖 2-9：非對稱式密碼系統
資料來源：[11]

非對稱式密碼系統的加密金鑰可以公開，傳送者利用接收者的公開加密金鑰來對明文加密，而接收者的解密金鑰必須妥善保存，要從非對稱式密碼系統的演算法、公開加密金鑰或是加密後的密文，推算出解密金鑰是不可能的[15][11]。目前常見的演算法有 RSA、ElGamal 等。

2.4.5 RSA 公開金鑰密碼系統

1978 年，美國麻省理工學院三位教授 R. Rivest、A. Shamir、A. Adleman 三位學者共同提出一個非對稱式的加密演算法。取他們三人姓氏的第一個英文字母，命名為 RSA 演算法[5]。

RSA 密碼系統用來對資料加密的公開金鑰是兩個正整數(e,N)，解密金鑰則是一個

數字 d 。公開金鑰與私密金鑰的產生方式如下：

步驟 1：先挑選兩個大質數 p 跟 q ，計算 $N = p \times q$ 。

步驟 2：再挑選一個正整數 e 作為加密金鑰， e 的值必須小於等於 $(p-1)\times(q-1)$ ，且 e 跟 $(p-1)\times(q-1)$ 必須互質，即 $GCD(e,(p-1)\times(q-1))=1$ 。

步驟 3：根據公式 $d \times e \mod (p-1)\times(q-1)=1$ ，計算正整數 d ， d 的值須小於等於 $(p-1)\times(q-1)$ ， d 即為私密解密金鑰。

加密金鑰 (e,N) 可以公開，由於從 p,q,e 可以算出解密金鑰 d ，所以 (p,q,d) 皆要妥善保密。當傳送者要將明文加密，傳送給接收者時，加密、解密的步驟如下：

步驟 1：接收者公開他的加密金鑰 (e,N) ，傳送者取得接收者的加密金鑰。

步驟 2：傳送者先將明文轉換成一個數字 M ， M 的值必須介於 0 到 N 之間，再計算 $M^e \bmod N$ ，得到的值即為密文 C ，傳送者將密文 C 傳送給接收者。

步驟 3：接收者收到密文 C 後，使用自己的私密解密金鑰 d ，計算 $C^d \bmod N$ ，即可得到原先的明文 M 。

給定兩個大質數 p 跟 q ，利用計算機的運算，我們可以很輕易的求出其乘積 N ；相對地，給定乘積 N ，要對 N 作質因數分解，求得原先相乘的兩個大質數 p 跟 q ，在數學上卻是一個很困難的問題。RSA 公開金鑰密碼系統的安全性建立在大數分解的困難度，在使用 RSA 密碼系統時，必須謹慎的選擇大質數 p 跟 q ，計算公開金鑰 N ，使得 N 公開後，任何人皆無法將 N 分解得到 p 跟 q ，進而求出解密金鑰 d ，目前常見的 RSA 公開金鑰長度在 1024 位元以上[15][22]。

2.4.6 單向雜湊函數(One-Way Hash Function)

單向雜湊函數(One-way hash function)是獨立於加解密系統以外的一種密碼系統，它主要用在於需要比對檔案內容是否經過修改上，因此，常用在明文認證(Authentication)或數位簽章(Digital Signature)上。在運作上，主要是將任意長度的訊息 M ，經過單向雜湊函數(One-Way Hash Function)的處理運算後，可以得到一個固定長度的雜湊函數值 $h(M)$ ，這個雜湊函數值我們又稱為訊息摘要(Message Digest)演算法或訊息指紋(Message Fingerprint)演算法，如圖 2-10 所示。而要從雜湊函數的輸出 $h(M)$ ，求得原本的輸入訊息 M ，在計算上是不可能做到的，所以，通常來說單向雜湊函數必須具備碰撞抵抗性(Collision Resistance)，亦即對一個訊息 M_1 ，要找到另一個不同的訊息 M_2 ，經過運算得到相同的雜湊函數值，即 $h(M_1)=h(M_2)$ ，在計算上是不可能的，亦即給定任意兩個不同但是大小相同的訊息 M_1 與 M_2 ，再經由經過運算得到的雜湊函數值 $h(M_1)$ 與 $h(M_2)$ 將不會相等[22][26]。

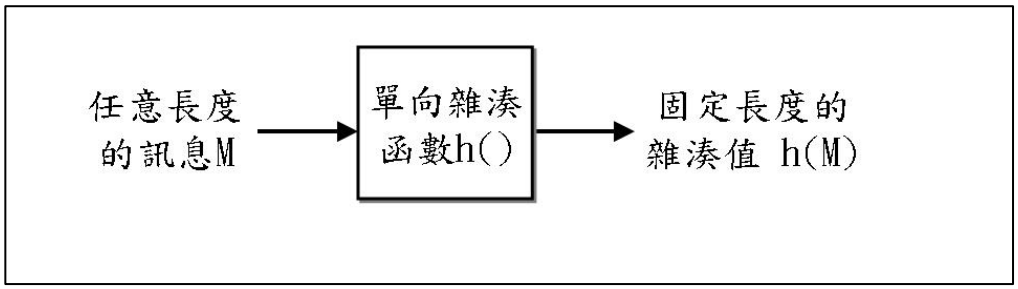

圖 2-10：單向雜湊函數
資料來源：[22]

目前常見的單向雜湊函數有麻省理工學院 R.Rivest 教授設計的 MD5 雜湊函數，以及美國國家標準科技局(NIST)的 SHA-1、SHA-256、SHA-384、SHA-512 等[22]。

2.5 Diffie-Hellman 公開金鑰交換演算法

1976 年，美國兩位學者 Diffie 與 Hellman[8]提出公開金鑰交換演算法，可以讓從未見面過的雙方，取得彼此的共通金鑰，作為雙方未來通訊或是資料傳輸加密的鑰匙。在進行金鑰交換通訊之前，通訊的雙方必須先選擇一個大質數 p 及 p 的原根 g，而且這個 g 必須滿足 $g < p$ 且 $g \neq 1$。g 是 p 的原根(primitive root)，表示 g 的 1 到 $p-1$ 不同次方被 p 除，餘數會產生 1 到 $p-1$ 的相異整數，但順序可能不同。亦即 $\{g^1 \bmod p, g^2 \bmod p, g^3 \bmod p, \cdots, g^{p-1} \bmod p\} = \{1,2,3,\cdots, p-1\}$。

接下來，我們簡單的介紹一下 Diffie 與 Hellman 的公開金鑰交換演算法是如何運作的，假設使用者 A 要與使用者 B 要進行通訊時，他們進行以下的步驟：

步驟 1：使用者 A 選擇一個隨機亂數 x，計算 $g^x \bmod p$，並且將 $g^x \bmod p$ 傳送給使用者 B，x 即為使用者 A 的私密金鑰。

步驟 2：使用者 B 選擇一個隨機亂數 y，計算 $g^y \bmod p$，並且將 $g^y \bmod p$ 傳送給使用者 A，y 即為使用者 B 的私密金鑰。

步驟 3：使用者 A 收到 B 傳送過來的 $g^y \bmod p$ 後，以自己的私密金鑰 x，計算 A、B 間的共通金鑰 $K_A = (g^y)^x \bmod p = g^{xy} \bmod p$。

步驟 4：使用者 B 收到 A 傳送過來的 $g^x \bmod p$ 後，以自己的私密金鑰 y，計算 A、B 間的共通金鑰 $K_B = (g^x)^y \bmod p = g^{xy} \bmod p$。

步驟 5：$K_A = K_B = g^{xy} \bmod p$ 為使用者 A、B 的共通金鑰[22]。

Diffie-Hellman 公開金鑰交換演算法的流程如圖 2-11 所示。

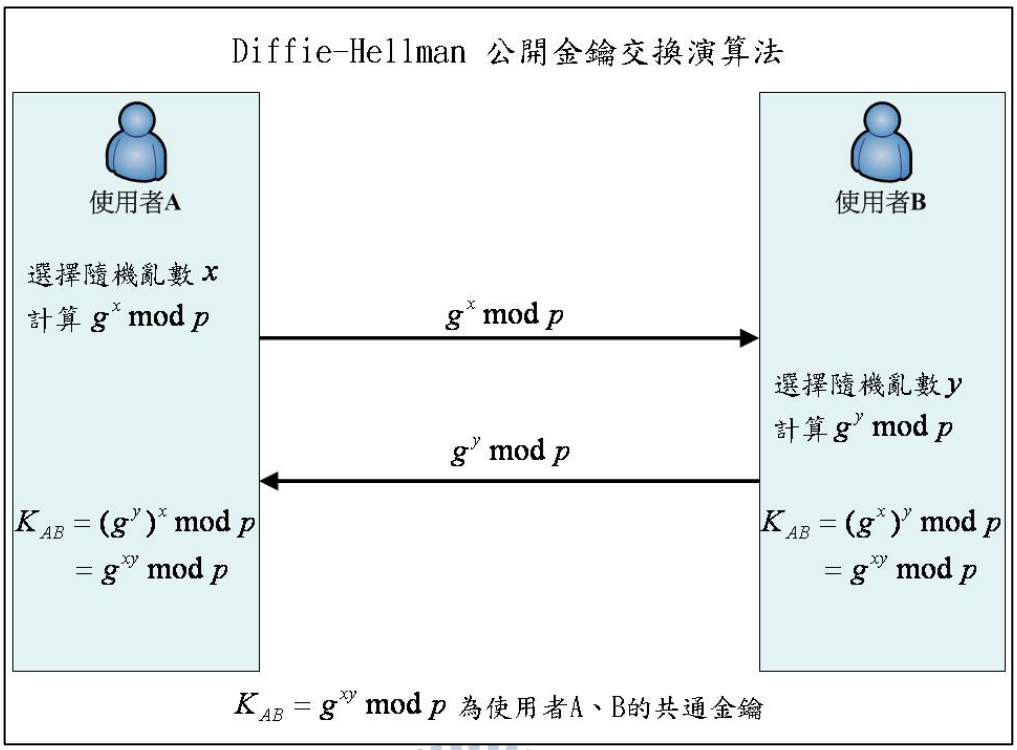

圖 2-11：Diffie-Hellman 公開金鑰交換演算法
資料來源：[22]

Diffie-Hellman 公開金鑰交換演算法的安全性主要架構在解離散對數(discrete logarithm)問題的困難性，因此，就算攻擊者即使可以在網路上擷取到大質數 p 、 p 的原根 g 、 g^x 與 g^y ，也很難推算出正確的 x,y 值，在計算上是不可行的。不過，Diffie-Hellman 公開金鑰交換演算法有著中間人攻擊的安全性問題，如圖 2-12 所示。

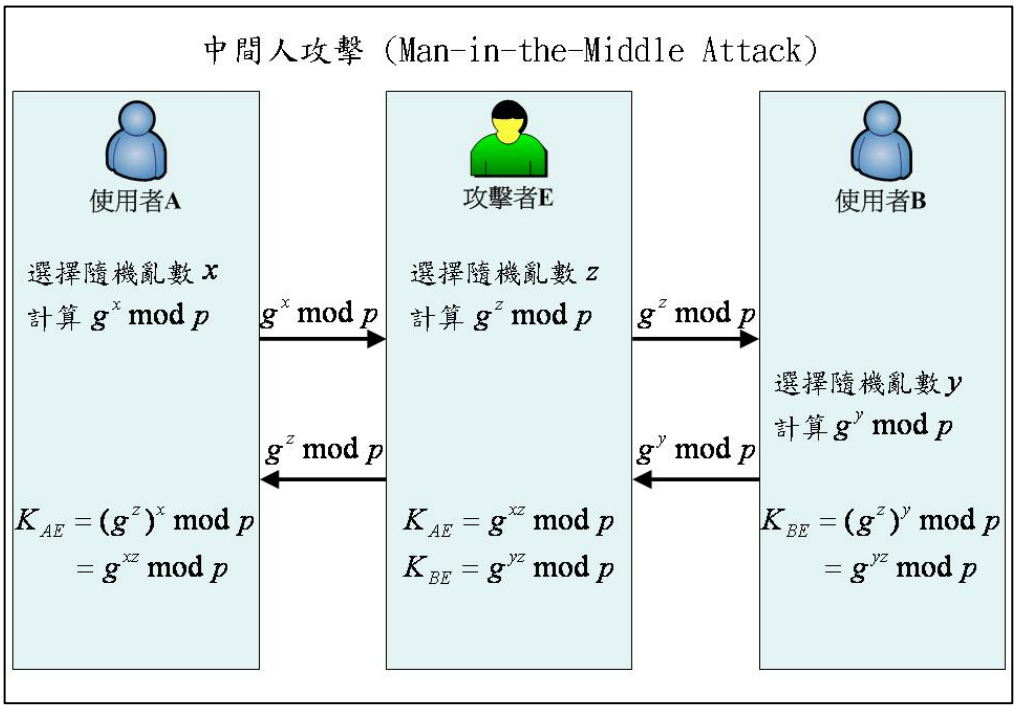

圖 2-12：中間人攻擊
資料來源：[22]

攻擊者 E 執行以下的步驟，即可攔截通訊雙方的訊息，並且讓使用者 A、B 誤以為自己是在跟對方通訊。

步驟 1：使用者 A 選擇隨機亂數 x，計算 $g^x \bmod p$，將 $g^x \bmod p$ 傳送給使用者 B。

步驟 2：攻擊者 E 攔截 $g^x \bmod p$，選擇隨機亂數 z，計算 $g^z \bmod p$，將 $g^z \bmod p$ 傳送給使用者 B。

步驟 3：使用者 B 選擇隨機亂數 y，計算 $g^y \bmod p$，將 $g^y \bmod p$ 傳送給使用者 A。

步驟 4：攻擊者 E 攔截 $g^y \bmod p$，將步驟 2 計算的 $g^z \bmod p$ 傳送給使用者 A。

步驟 5：使用者 A 收到 $g^z \bmod p$ 後，計算共通金鑰 $K_{AE} = (g^z)^x \bmod p = g^{xz} \bmod p$。

步驟 6：使用者 B 收到 $g^z \bmod p$ 後，計算共通金鑰 $K_{BE} = (g^z)^y \bmod p = g^{yz} \bmod p$

步驟 7：攻擊者 E 計算 $K_{AE} = (g^x)^z \bmod p = g^{xz} \bmod p$ 與

$K_{BE} = (g^y)^z \bmod p = g^{yz} \bmod p$，即可輕易得知使用者 A、B 通訊的內容。

很明顯地，Diffie-Hellman 公開金鑰交換演算法遭受中間人攻擊的主要原因，在於使用者 A、B 無法確認對方的身份。要避免這種類型的攻擊，必須對通訊的雙方進行身份驗證[22]。

2.6 學者 Lee 等人提出的三方驗證金鑰交換協定

對公開的網路環境而言，資料要能安全的傳輸，必須仰賴密碼學的加密技術。如何讓網路通訊的雙方，驗證彼此的身份(Authentication)，並且交換加密金鑰(Encrypted Key Exchange，EKE)，便成為網路安全一項很重要的議題。

學者 Yeh[1]等人為了讓通訊的雙方可以藉由一個公正的伺服器進行通訊，並且避免密碼猜測攻擊，在 2003 年提出一個有效率的三方驗證金鑰協定。2009 年，學者 Lee[7]等人對學者 Yeh 等人的協定提出了一個改善效率的方法。在這裡先簡要地介紹學者 Lee 等人所提出的三方驗證金鑰交換協定。在這個協定中，所使用的符號說明如下表：

表 2-1：學者 Lee 等人所提出的三方驗證金鑰交換協定符號說明表

符號	說　　明
ID_A, ID_B, S	ID_A 與 ID_B 為使用者 A、B 的帳號，S 為可信賴的伺服器
PW_A, PW_B	使用者 A, B 與伺服器 S 分享的密碼
K_S	可信賴伺服器 S 的公開金鑰
x, y, r_A, r_B	隨機亂數
K	使用者 A, B 的階段性金鑰
$\{message\}_K$	以非對稱式金鑰 K 對"message"加密
$[message]_K$	以對稱式金鑰 K 對"message"加密
$h()$	單向雜湊函數
p	一個非常大的質數
g	p 的原根(primitive root)
$A \rightarrow B : M$	表示從 A 傳送訊息 M 給 B
Fa	由 B 產生的訊息，用來確認階段性金鑰的正確性

資料來源：[7]

學者 Lee 等人提出的三方驗證金鑰交換協定，如圖 2-13 所示。

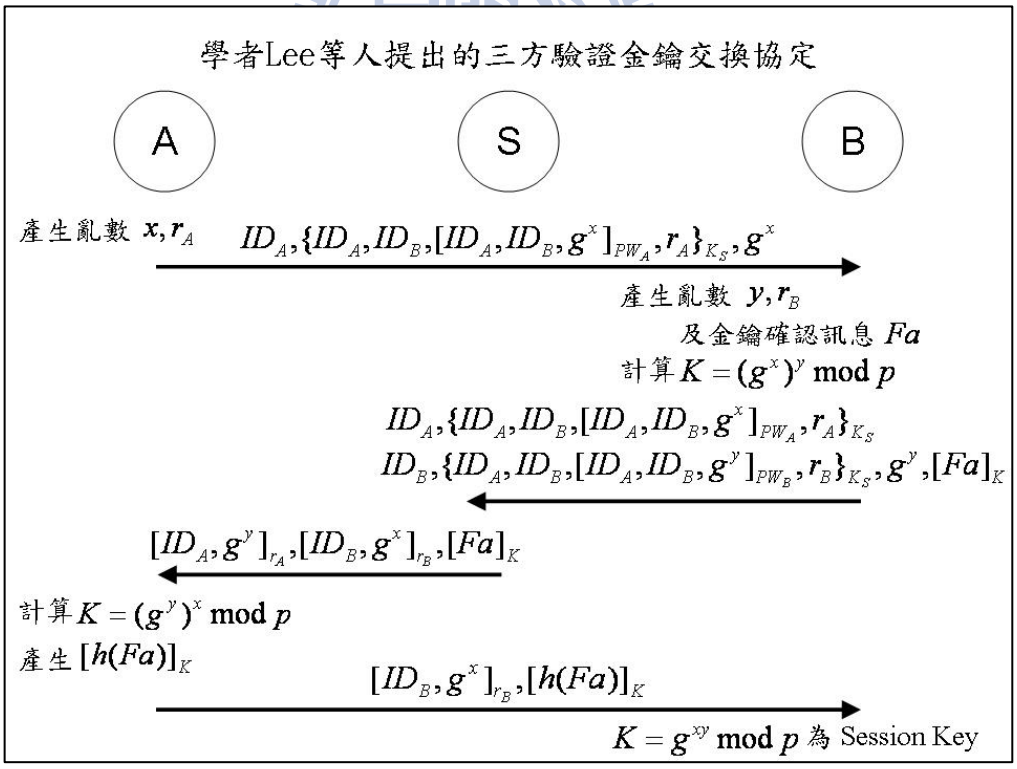

圖 2-13：學者 Lee 等人提出的三方驗證金鑰交換協定

資料來源：[7]

使用者 A、B 透過可信賴的伺服器 S 驗證彼此的身份，並且建立共通金鑰的過程，

如以下的步驟：

步驟 1：使用者 A 挑選兩個隨機亂數 r_A, x，計算 $g^x \pmod p$。以自己的密碼 PW_A 為私密金鑰，採用對稱式密碼系統對使用者本身的帳號 ID_A、要進行通訊的使用者 B 的帳號 ID_B，以及 g^x 加密，得到 $[ID_A, ID_B, g^x]_{PW_A}$；再以伺服器 S 的公開金鑰 K_S，採用非對稱式密碼系統對 $ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A$ 加密，得到 $\{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}$。

步驟 2：A → B： $ID_A, \{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}, g^x$

使用者 A 將本身的帳號 ID_A、步驟 1 加密得到的 $\{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}$，以及 g^x 傳送給使用者 B。

步驟 3：使用者 B 收到使用者 A 傳送過來的訊息後，儲存 g^x。

挑選兩個隨機亂數 r_B, y，與使用者 A 在步驟 1 的計算加密流程相同，產生一個 $\{ID_A, ID_B, [ID_A, ID_B, g^y]_{PW_B}, r_B\}_{K_S}$ 的訊息。

計算共通金鑰 $K = (g^x)^y \bmod p = g^{xy} \bmod p$，產生一個金鑰確認訊息 Fa，並以 K 加密得到 $[Fa]_K$。

步驟 4：B → S： $ID_A, \{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}$

$ID_B, \{ID_A, ID_B, [ID_A, ID_B, g^y]_{PW_B}, r_B\}_{K_S}, g^y, [Fa]_K$

使用者 B 將使用者 A 傳送過來的 $ID_A, \{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}$ 與 $ID_B, \{ID_A, ID_B, [ID_A, ID_B, g^y]_{PW_B}, r_B\}_{K_S}, g^y, [Fa]_K$ 傳送給伺服器 S。

步驟 5：伺服器 S 對 $\{ID_A, ID_B, [ID_A, ID_B, g^x]_{PW_A}, r_A\}_{K_S}$ 與 $\{ID_A, ID_B, [ID_A, ID_B, g^y]_{PW_B}, r_B\}_{K_S}$ 進行解密，再以使用者 A、B 的密碼 PW_A、PW_B 分別對 $[ID_A, ID_B, g^x]_{PW_A}$ 與 $[ID_A, ID_B, g^y]_{PW_B}$ 解密，驗證使用者 A、B 的身份。驗證完成後，以 r_A, r_B 分別對 ID_A, g^y 與 ID_B, g^x 加密，產生 $[ID_A, g^y]_{r_A}$、$[ID_B, g^x]_{r_B}$。

步驟 6：S → A： $[ID_A, g^y]_{r_A}, [ID_B, g^x]_{r_B}, [Fa]_K$

伺服器 S 將 $[ID_A, g^y]_{r_A}, [ID_B, g^x]_{r_B}$ 以及 $[Fa]_K$ 傳送給使用者 A。

步驟 7：使用者 A 以 r_A 對 $[ID_A, g^y]_{r_A}$ 解密，驗證伺服器 S 的身份，並計算共通金鑰 $K = (g^y)^x \bmod p = g^{xy} \bmod p$。使用 K 對 $[Fa]_K$ 解密，得到 Fa，計算 $h(Fa)$，對 $h(Fa)$ 加密得到 $[h(Fa)]_K$。

步驟 8：A → B： $[ID_B, g^x]_{r_B}, [h(Fa)]_K$

使用者 A 將 $[ID_B, g^x]_{r_B}, [h(Fa)]_K$ 傳送給使用者 B。

步驟 9：使用者 B 以 r_B 對 $[ID_B, g^x]_{r_B}$ 解密，驗證伺服器 S 的身份與 g^x 的正確性。使用在步驟 3 計算的共通金鑰 K，對 $[h(Fa)]_K$ 解密得到 $h(Fa)$，檢驗 $h(Fa)$ 的正確性，確認使用者 A、B 持有相同的共通金鑰 K。

第三章 即時通訊協定設計

本研究設計的通訊協定有三個階段：

1. 註冊期(Registration Phase)

使用者向即時通訊伺服器申請，註冊成為合法的使用者。

2. 登入期(Login Phase)

伺服器將非對稱式公開金鑰K_S、大質數p與p的原根g存於 Applet 程式中，傳送 Applet 程式給使用者，使用者傳送帳號登入即時通訊伺服器。

3. 三方驗證與使用者加密金鑰交換

(Three Party Authentication and Encrypted Key Exchange)

要進行交談的兩位使用者，透過即時通訊伺服器驗證身分，並建立彼此通訊的階段性共通金鑰。

在本研究的通訊協定中，所使用的符號如下表：

表 3-1：本研究設計的通訊協定符號說明表

符號	說　明
ID_A, ID_B, S	ID_A與ID_B為使用者 A, B 的帳號，S 為可信賴的伺服器
PW_A, PW_B	使用者 A, B 與即時通訊伺服器 S 儲存的密碼
K_S, k_S	伺服器 S 的非對稱式公開金鑰與私密金鑰
x, y, r_A, r_B, r_S	隨機產生的亂數
K_A, K_B	使用者 A, B 各自計算的階段性金鑰
$\{message\}_K$	使用非對稱式公開金鑰K對"message"加密
$[message]_K$	使用對稱式金鑰K對"message"加密
$h()$	單向雜湊函數
$\|\|$	參數連結
p	大質數
g	大質數p的原根(primitive root)
$A \rightarrow B : M$	表示從 A 傳送訊息 M 給 B

3.1 註冊期(Registration Phase)

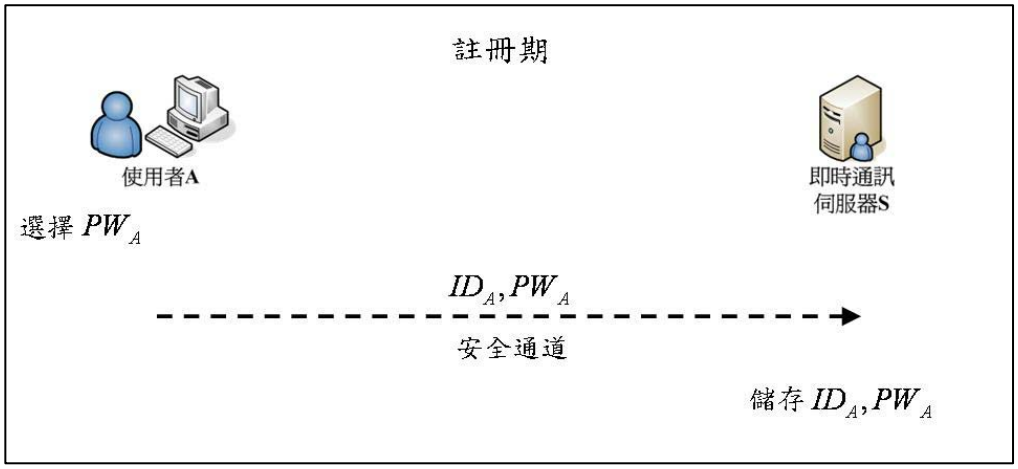

圖 3-1：本研究設計的通訊協定註冊期

當使用者 A 要使用即時通訊的服務時，必須先向即時通訊伺服器 S 提出申請，註冊成為一個合法的使用者。使用者 A 先挑選一組帳號 ID_A 及密碼 PW_A，隨後將帳號 ID_A 與密碼 PW_A，透過安全通道的方式，傳送給即時通訊伺服器 S，整個註冊期的流程，如下面的步驟所示：

步驟 1：使用者 A 挑選一組帳號 ID_A 及密碼 PW_A。

步驟 2：A → S： ID_A, PW_A

使用者 A 將帳號 ID_A 與密碼 PW_A，透過安全通道，傳送給即時通訊伺服器 S。

步驟 3：伺服器 S 在收到使用者 A 申請的註冊帳號 ID_A 與密碼 PW_A 後，儲存使用者的帳號與密碼。

3.2 登入期(Login Phase)

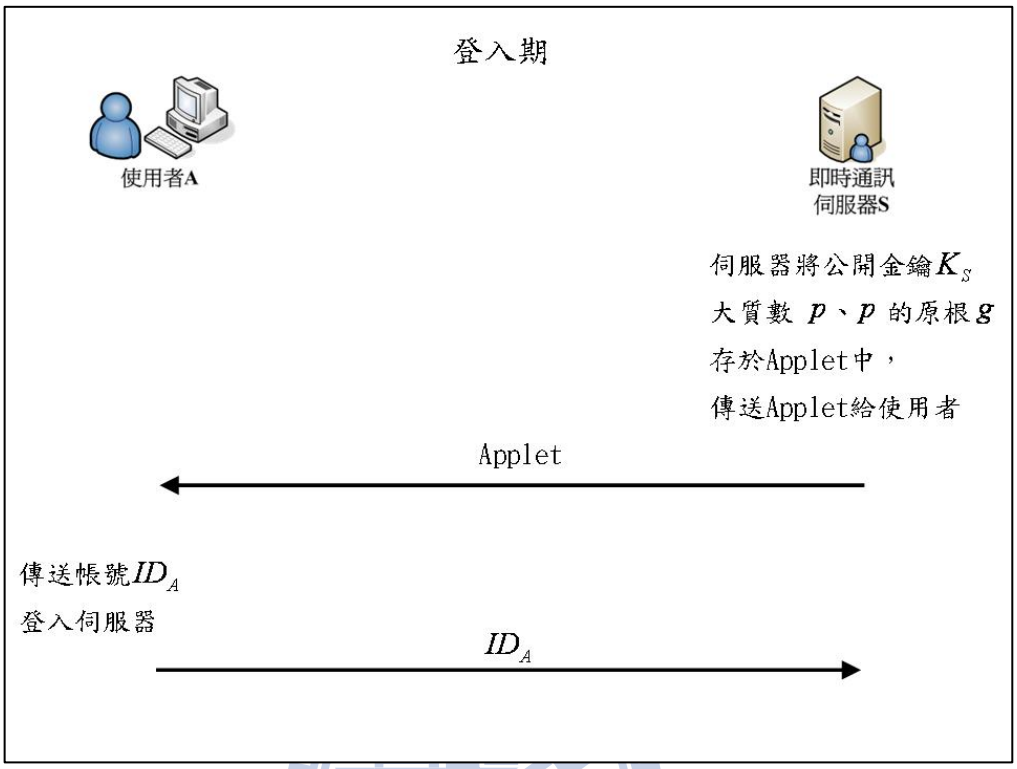

圖 3-2：本研究設計的通訊協定登入期

當一個已經註冊的合法使用者 A 想要登入即時通訊伺服器 S，使用即時通訊的服務時，必須執行下面的登入步驟：

步驟 1：S → A：Applet

即時通訊伺服器 S 將伺服器的非對稱式公開金鑰K_S、大質數 p、p 的原根 g 存於 Applet 程式中，將 Applet 程式傳送給使用者 A。

步驟 2：A → S：ID_A

使用者 A 將註冊的帳號ID_A傳送給即時通訊伺服器 S。

3.3 三方驗證與使用者加密金鑰交換(Three Party Authentication and Encrypted Key Exchange)

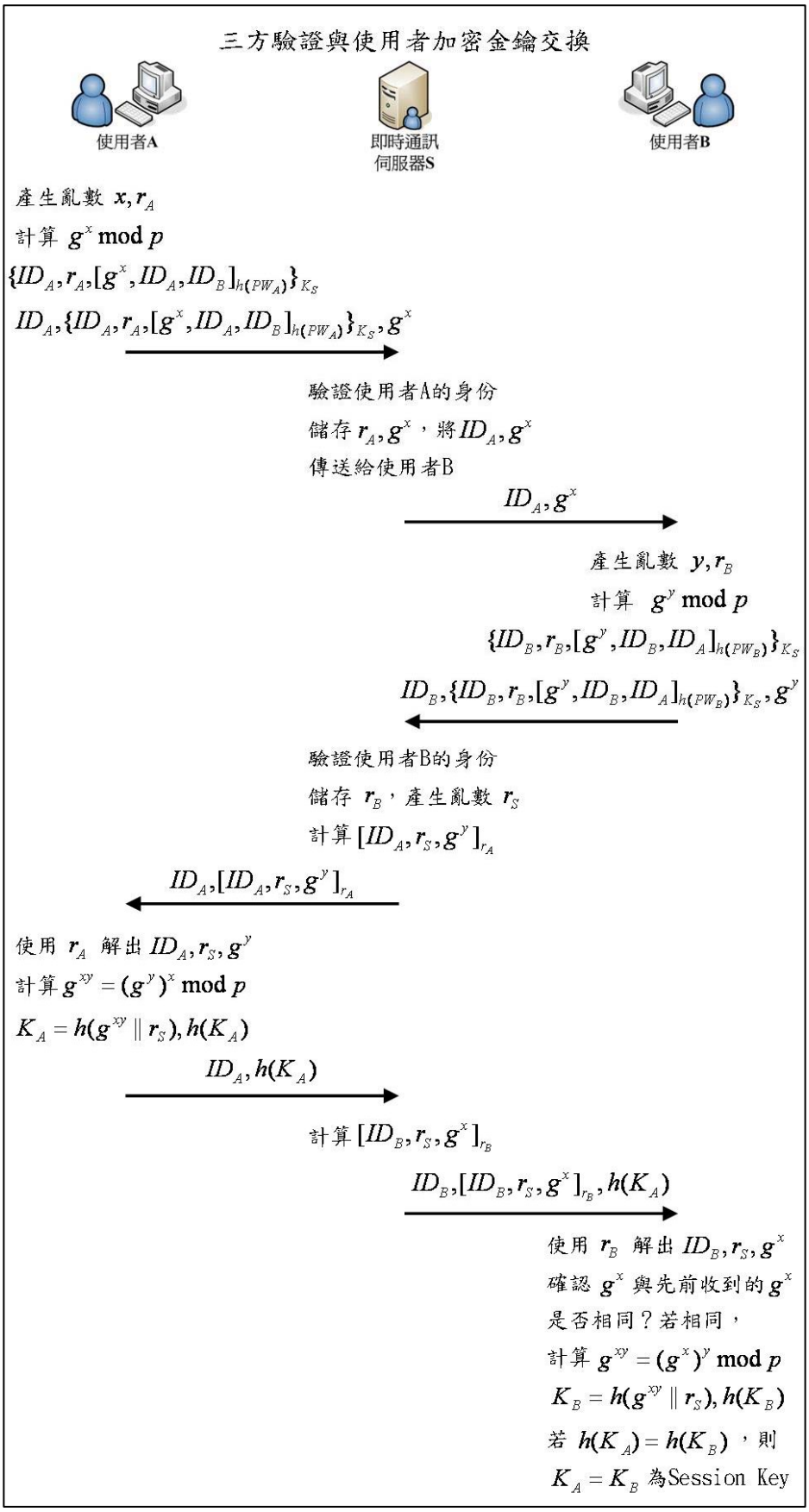

圖 3-3：三方驗證與使用者加密金鑰交換

在合法使用者登入即時通訊伺服器 S 後，若使用者 A 要與使用者 B 交談進行通訊，則使用者 A、B 之間必須事先建立一把階段性的共通金鑰(Session Key)，並且將即時通訊的內容利用這一把共通金鑰加密，再傳送到網路上。網路上的攻擊者即使可以擷取使用者 A、B 之間所有傳輸通訊的訊息，如果沒有 A、B 交談的共通金鑰，便無法還原這些訊息，得知通訊的內容。使用者 A、B 透過可信賴的即時通訊伺服器 S 驗證彼此的身份，並且建立共通金鑰的過程，如以下的步驟：

步驟 1：使用者 A 挑選兩個隨機亂數 x, r_A，計算 $g^x \bmod p$。以密碼的單向雜湊函數值 $h(PW_A)$ 為對稱式金鑰，對使用者 A 本身的帳號 ID_A、要進行通訊的使用者 B 的帳號 ID_B，以及 g^x 加密，得到 $[g^x, ID_A, ID_B]_{h(PW_A)}$；再以即時通訊伺服器 S 的非對稱式公開金鑰 K_S，對 $ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}$ 加密，得到 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$。

步驟 2：A → S： $ID_A, \{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}, g^x$

使用者 A 將本身的帳號 ID_A、步驟 1 加密得到的 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$，以及 g^x 傳送給伺服器 S。

步驟 3：伺服器 S 在收到使用者 A 傳送過來的 $ID_A, \{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}, g^x$ 後，以伺服器 S 的非對稱式私密金鑰 k_S 對 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$ 解密，得到 $ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}$。再計算儲存在伺服器的密碼 PW_A 的單向雜湊函數值 $h(PW_A)$，以 $h(PW_A)$ 為對稱式金鑰對 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 解密，得到 g^x, ID_A, ID_B。

將使用者 A 傳送過來的帳號 ID_A、伺服器 S 以非對稱式私密金鑰 k_S 解密所得到的帳號 ID_A，以及 $h(PW_A)$ 為對稱式金鑰解密所得到的帳號 ID_A 做比對，根據這三個帳號是否完全相同，來驗證使用者 A 的身份。檢驗帳號的過程，若有任何一個帳號不同，立即中斷使用者 A 的連線。

在使用者 A 的身份確認無誤後，將使用者 A 直接傳送過來的 g^x，與 $h(PW_A)$ 對 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 解密得到的 g^x 做比對，檢驗 g^x 的正確性。若不相等，則中斷與使用者 A 的連線。在驗證使用者 A 的身份與正確的 g^x 後，伺服器 S 儲存 r_A, g^x 的值。

步驟 4：S → B： ID_A, g^x

伺服器 S 將使用者 A 的帳號 ID_A 與 g^x 傳送給使用者 B。

步驟 5：使用者 B 在收到伺服器 S 傳送過來的 ID_A, g^x 後，儲存 g^x。挑選兩個隨機亂數 y, r_B，計算 $g^y \bmod p$。以密碼的單向雜湊函數值 $h(PW_B)$ 為對稱式金鑰，對使用者 B 本身的帳號 ID_B、要進行通訊的使用者 A 的帳號 ID_A，以及 g^y 加密，得到 $[g^y, ID_B, ID_A]_{h(PW_B)}$；再以即時通訊伺服器 S 的非對稱式公開金鑰

K_S，對 $ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}$ 加密，得到 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$。

步驟 6：B → S： $ID_B, \{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}, g^y$

使用者 B 將本身的帳號 ID_B、步驟 5 加密得到的 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$，以及 g^y 傳送給伺服器 S。

在步驟 1 與步驟 5 的過程中，使用者 A、B 以 $h(PW_A)$、$h(PW_B)$ 對 g^x, ID_A, ID_B 與 g^y, ID_B, ID_A 加密，得到 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B)}$，除了將 g^x 與 g^y 安全的傳送給伺服器 S，並且讓伺服器 S 可以藉由使用者 A、B 的密碼，來驗證要進行通訊雙方的身份。

使用者 A、B 再以伺服器 S 的非對稱式公開金鑰 K_S 對 $ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}$ 與 $ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}$ 加密，得到 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$ 與 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$，除了將 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B)}$ 安全的傳送給伺服器 S，並將隨機亂數 r_A 與 r_B 傳送給伺服器 S，讓後續伺服器 S 可以利用 r_A、r_B 對訊息加密，傳遞給使用者 A、B，使用者 A、B 也可以藉由 r_A、r_B 來驗證伺服器 S 的身份。

步驟 7：伺服器 S 在收到使用者 B 傳送過來的 $ID_B, \{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}, g^y$ 後，以伺服器 S 的非對稱式私密金鑰 k_S 對 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$ 解密，得到 $ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}$。再計算儲存在伺服器的密碼 PW_B 的單向雜湊函數值 $h(PW_B)$，以 $h(PW_B)$ 為對稱式金鑰對 $[g^y, ID_B, ID_A]_{h(PW_B)}$ 解密，得到 g^y, ID_B, ID_A。

將使用者 B 傳送過來的帳號 ID_B、伺服器 S 以非對稱式私密金鑰 k_S 解密所得到的帳號 ID_B，以及 $h(PW_B)$ 為對稱式金鑰解密所得到的帳號 ID_B 做比對，根據這三個帳號是否完全相同，來驗證使用者 B 的身份。檢驗帳號的過程，若有任何一個帳號不同，立即中斷使用者 B 的連線。

在使用者 B 的身份確認無誤後，將使用者 B 直接傳送過來的 g^y，與 $h(PW_B)$ 對 $[g^y, ID_B, ID_A]_{h(PW_B)}$ 解密得到的 g^y 做比對，檢驗 g^y 的正確性。若不相等，則中斷與使用者 B 的連線。

在驗證使用者 B 的身份與正確的 g^y 後，伺服器 S 儲存 r_B 的值，挑選隨機亂數 r_S，以步驟 3 儲存的 r_A 為對稱式金鑰，對 ID_A, r_S, g^y 加密，得到 $[ID_A, r_S, g^y]_{r_A}$。

步驟 8：S → A： $ID_A, [ID_A, r_S, g^y]_{r_A}$

伺服器 S 將使用者 A 的帳號 ID_A，與步驟 7 加密得到的 $[ID_A, r_S, g^y]_{r_A}$ 傳送給使用者 A。

在步驟 3 與步驟 7 的過程中，伺服器 S 最主要的任務在驗證使用者 A、B 的

身份，並讓使用者 A、B 各自產生的 g^x 與 g^y，能夠透過伺服器 S 進行交換，再安全的傳送給使用者 B、A，以便後續通訊的雙方可以產生階段性共通金鑰。

步驟 9：使用者 A 在收到伺服器 S 傳送過來的 $ID_A,[ID_A,r_S,g^y]_{r_A}$ 後，使用步驟 1 挑選的隨機亂數 r_A 為對稱式金鑰，對 $[ID_A,r_S,g^y]_{r_A}$ 進行解密，得到 ID_A,r_S,g^y，根據解密得到的帳號 ID_A 是否為使用者 A 本身的帳號，來驗證伺服器 S 的身份，若不正確，則中斷與伺服器 S 的連線。

在驗證伺服器 S 的身份後，計算 $(g^y)^x \bmod p = g^{xy} \bmod p$，產生階段性的共通金鑰 $K_A = h(g^{xy}, r_S)$ 與金鑰的單向雜湊函數值 $h(K_A)$。

步驟 10：A → S： $ID_A, h(K_A)$

使用者 A 將本身的帳號 ID_A 與共通金鑰的單向雜湊函數值 $h(K_A)$，傳送給伺服器 S。

步驟 11：伺服器 S 在收到使用者 A 傳送過來的 $ID_A, h(K_A)$ 後，以步驟 7 儲存的 r_B 為對稱式金鑰，對 ID_B, r_S, g^x 加密，得到 $[ID_B, r_S, g^x]_{r_B}$。

步驟 12：S → B： $ID_B,[ID_B,r_S,g^x]_{r_B},h(K_A)$

伺服器 S 將使用者 B 的帳號 ID_B、步驟 11 加密得到的 $[ID_B,r_S,g^x]_{r_B}$ 與收到的 $h(K_A)$，傳送給使用者 B。

步驟 13：使用者 B 在收到伺服器 S 傳送過來的 $ID_B,[ID_B,r_S,g^x]_{r_B},h(K_A)$ 後，使用步驟 5 挑選的隨機亂數 r_B 為對稱式金鑰，對 $[ID_B,r_S,g^x]_{r_B}$ 進行解密，得到 ID_B,r_S,g^x，根據解密得到的帳號 ID_B 是否為使用者 B 本身的帳號，來驗證伺服器 S 的身份，若不正確，則中斷與伺服器 S 的連線。

在驗證伺服器 S 的身份後，將步驟 5 儲存的 g^x，與 r_B 對 $[ID_B,r_S,g^x]_{r_B}$ 解密得到的 g^x 做比對，檢驗 g^x 的正確性。若不相等，則中斷與伺服器 S 的連線。

在驗證伺服器 S 的身份與正確的 g^x 後，計算 $(g^x)^y \bmod p = g^{xy} \bmod p$，產生階段性的共通金鑰 $K_B = h(g^{xy}, r_S)$ 與金鑰的單向雜湊函數值 $h(K_B)$。

比對 $h(K_A)$ 與 $h(K_B)$，若使用者 A、B 各自計算金鑰的單向雜湊函數值相等，即 $h(K_A) = h(K_B)$，則 $K_A = K_B$，使用者 A、B 持有相同的階段性共通金鑰。若 $h(K_A) \neq h(K_B)$，則 $K_A \neq K_B$，使用者 A、B 各自計算的階段性共通金鑰並不相同，g^x, g^y, r_A, r_B, r_S 在驗證的過程中可能遭到惡意竄改，立即中斷與伺服器 S 的連線。

第四章 安全性分析與效能分析

本章就本研究設計的即時通訊協定做安全性分析與效能分析。

4.1 雙向驗證(Mutual Authentication)

在通訊協定的步驟 3，即時通訊伺服器 S 在收到使用者 A 傳送過來的 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$ 後，以伺服器的非對稱式私密金鑰 k_S 解密，得到 $ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}$，再計算密碼的單向雜湊函數值 $h(PW_A)$，對 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 解密，得到 g^x, ID_A, ID_B。將使用者 A 傳送過來的帳號 ID_A、伺服器 S 以非對稱式私密金鑰 k_S 解密所得到的帳號 ID_A，以及 $h(PW_A)$ 為對稱式金鑰解密所得到的帳號 ID_A 做比對，根據這三個帳號是否完全相同，來驗證使用者 A 的身份。

同樣地，在通訊協定的步驟 7，即時通訊伺服器 S 在收到使用者 B 傳送過來的 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$ 後，以伺服器的非對稱式私密金鑰 k_S 解密，得到 $ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}$，再計算密碼的單向雜湊函數值 $h(PW_B)$，對 $[g^y, ID_B, ID_A]_{h(PW_B)}$ 解密，得到 g^y, ID_B, ID_A。將使用者 B 傳送過來的帳號 ID_B、伺服器 S 以非對稱式私密金鑰 k_S 解密所得到的帳號 ID_B，以及 $h(PW_B)$ 為對稱式金鑰解密所得到的帳號 ID_B 做比對，根據這三個帳號是否完全相同，來驗證使用者 B 的身份。

而在步驟 9，使用者 A 在收到伺服器 S 傳送過來的 $[ID_A, r_S, g^y]_{r_A}$ 後，使用步驟 1 挑選的隨機亂數 r_A，對 $[ID_A, r_S, g^y]_{r_A}$ 進行解密，得到 ID_A, r_S, g^y，根據解密得到的帳號 ID_A 是否為使用者 A 本身的帳號，來驗證伺服器 S 的身份。

同樣地，在步驟 13，使用者 B 在收到伺服器 S 傳送過來的 $[ID_B, r_S, g^x]_{r_B}$ 後，使用步驟 5 挑選的隨機亂數 r_B 對 $[ID_B, r_S, g^x]_{r_B}$ 進行解密，得到 ID_B, r_S, g^x，根據解密得到的帳號 ID_B 是否為使用者 B 本身的帳號，來驗證伺服器 S 的身份。由以上可知，我們的通訊協定可以雙向驗證伺服器 S 與使用者 A、B 之間的身份。

4.2 中間人攻擊(Man-in-the-Middle Attack)

假設攻擊者在網路上擷取到 g^x, g^y，並且選擇隨機亂數 z，計算 g^z 傳送給使用者 A、B，嘗試在使用者 A、B 之間進行中間人攻擊(Man-in-the-Middle Attack)。然而，在我們的通訊協定中，$K = h(g^{xy}, r_S)$ 為使用者 A、B 交談的階段性共通金鑰(Session Key)，r_S 是即時通訊伺服器 S 挑選的隨機亂數，在驗證的過程中以使用者 A、B 挑選的隨機亂數 r_A, r_B 加密傳送；而 r_A, r_B 在驗證的過程也以伺服器 S 的非對稱式公開金鑰 K_S 加密後再

傳送。攻擊者即使可以在網路上擷取到 g^x, g^y，根據解離散對數問題的困難性，也很難推算出正確的 x, y 值，更無法取得隨機亂數 r_A, r_B，進一步解出 r_S，計算出正確的階段性共通金鑰。

在通訊協定的步驟 13，使用者 B 藉由比對 $h(K_A)$ 與 $h(K_B)$，可以進一步的確認使用者 A、B 是否持有相同的階段性共通金鑰。若 $h(K_A) = h(K_B)$，則 $K_A = K_B = h(g^{xy}, r_S)$，表示通訊協定驗證的流程順利完成，使用者 A、B 持有相同的共通金鑰。若 $h(K_A) \neq h(K_B)$，表示使用者 A、B 各自計算的共通金鑰並不相同，即 $K_A \neq K_B$，g^x、g^y、r_A、r_B、r_S 在驗證的過程中可能遭到惡意竄改。在我們的通訊協定中，提供了伺服器 S 與使用者間的雙向驗證，使用者 A、B 藉由可信賴伺服器 S，可以相互驗證彼此的身份，因此可以有效的避免中間人攻擊(Man-in-the-Middle Attack)。

4.3 密碼猜測攻擊(Password Guessing Attack)

在通訊協定的步驟 3 與步驟 7，伺服器 S 以非對稱式私密金鑰 k_S 對 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$ 與 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$ 解密，得到 $ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}$ 與 $ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}$，再以密碼的單向雜湊函數值 $h(PW_A)$、$h(PW_B)$，分別對 $[g^x, ID_A, ID_B]_{h(PW_A)}$、$[g^y, ID_B, ID_A]_{h(PW_B)}$ 解密，得到 g^x, ID_A, ID_B 與 g^y, ID_B, ID_A。將使用者 A、B 傳送過來的帳號 ID_A、ID_B，伺服器 S 以非對稱式私密金鑰 k_S 解密所得到的帳號 ID_A、ID_B，以及 $h(PW_A)$、$h(PW_B)$ 為對稱式金鑰解密所得到的帳號 ID_A、ID_B 做比對，根據使用者 A、B 這三個帳號是否完全相同，來驗證身份，確認要進行通訊的雙方確實為使用者 A、B 無誤。

4.3.1 線上密碼猜測攻擊(On-Line Password Guessing Attack)

假設攻擊者 E 嘗試猜測使用者 A、B 的密碼 PW_A、PW_B，或密碼的單向雜湊函數值 $h(PW_A)$、$h(PW_B)$，以錯誤的 $h(PW_A')$、$h(PW_B')$ 對 g^x, ID_A, ID_B 與 g^y, ID_B, ID_A 加密，得到 $[g^x, ID_A, ID_B]_{h(PW_A')}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B')}$，並將加密過的訊息傳送給伺服器 S。在步驟 3 與步驟 7，當伺服器 S 以正確的 $h(PW_A)$、$h(PW_B)$ 對 $[g^x, ID_A, ID_B]_{h(PW_A')}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B')}$ 進行解密時，所得到的 ID_A', ID_B'，與使用者 A、B 原本的帳號 ID_A, ID_B 將不會一致，攻擊者嘗試猜測密碼的行為會被察覺，伺服器 S 會立即中斷攻擊者的連線。如果有攻擊者以錯誤的猜測密碼密集登入，企圖通過伺服器的驗證，可以在伺服器設立一個計數器，記錄使用者以錯誤密碼登入的次數，在一個固定時間內，驗證失敗的次數達到一定的安全次數後，伺服器將會停止攻擊者的驗證要求，因此我們的通訊協定可以

有效的避免線上密碼猜測攻擊(On-Line Guessing Attack)。

4.3.2 無法偵測的線上密碼猜測攻擊

(Undetectable On-Line Password Guessing Attack)

在我們的通訊協定中，提供了伺服器與使用者 A、B 之間的相互驗證，並且是由伺服器 S 先以 $h(PW_A)$ 、$h(PW_B)$ 驗證使用者 A、B 的身份後，再由使用者 A、B 驗證伺服器 S 的身份。假設攻擊者 E 以錯誤的 $h(PW_A')$ 、$h(PW_B')$ 對 g^x, ID_A, ID_B 與 g^y, ID_B, ID_A 加密，得到 $[g^x, ID_A, ID_B]_{h(PW_A')}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B')}$ ，並將加密過的訊息傳送給伺服器 S，嘗試藉由伺服器的回應，猜測出正確的 $h(PW_A)$ 或 $h(PW_B)$ 。很明顯的，在步驟 3 與步驟 7，當伺服器 S 以正確的 $h(PW_A)$ 、$h(PW_B)$ 對 $[g^x, ID_A, ID_B]_{h(PW_A')}$ 與 $[g^y, ID_B, ID_A]_{h(PW_B')}$ 進行解密時，所得到的 ID_A', ID_B' ，與使用者 A、B 原本的帳號 ID_A, ID_B 將不會一致，所以，伺服器 S 會立即中斷攻擊者的連線，對錯誤的密碼驗證不會給予任何回應訊息。在使用者身份驗證的過程中，攻擊者嘗試猜測密碼的行為一定會被伺服器察覺，在伺服器中斷使用者的連線後，攻擊者無法藉由伺服器回應的訊息，來檢驗猜測的密碼是否正確，因此，我們的通訊協定可以有效的避免無法偵測的線上密碼猜測攻擊(Undetectable On-Line Password Guessing Attack)。

4.3.3 離線密碼猜測攻擊(Off-Line Password Guessing Attack)

假設有攻擊者企圖在驗證的過程進行離線密碼猜測攻擊，那麼他至少必須取得 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 、$[g^y, ID_B, ID_A]_{h(PW_B)}$ 這兩個訊息才有可能進行。

在通訊協定的步驟 1 與步驟 5，使用者 A、B 在傳送 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 、$[g^y, ID_B, ID_A]_{h(PW_B)}$ 前，均以伺服器 S 的非對稱式公開金鑰 K_S 加密後，再傳送到網路上，攻擊者如果沒有伺服器 S 的非對稱式私密金鑰 k_S ，就無法對 $\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$ 與 $\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$ 進行解密，得到正確的 $[g^x, ID_A, ID_B]_{h(PW_A)}$ 、$[g^y, ID_B, ID_A]_{h(PW_B)}$ 。攻擊者無法直接從網路上，擷取到任何可以進行離線密碼猜測攻擊的訊息，因此我們的通訊協定可以有效的避免離線密碼猜測攻擊(Off-Line Guessing Attack)。

4.4 伺服器偽裝攻擊(Server Spoofing Attack)

假設有攻擊者偽裝成一部合法的伺服器，嘗試欺騙使用者登入，企圖竊取使用者的

驗證資料。那麼這一部伺服器必須要有伺服器 S 的非對稱式私密金鑰k_S，否則在通訊協定的步驟 3 與步驟 7，當偽裝伺服器要對使用者 A、B 傳送過來的$\{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}$與$\{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}$進行解密時，無法得到正確的$r_A$及$r_B$。假設攻擊者自行產生錯誤的隨機亂數$r_A'$,$r_B'$及$r_S'$，並且以偽造的$r_A'$,$r_B'$來對$ID_A, r_S', g^y$與$ID_B, r_S', g^x$加密；在步驟 9，當使用者 A 以正確的$r_A$及對偽裝伺服器傳送過來的$[ID_A, r_S', g^y]_{r_A'}$進行解密時，所得到的$ID_A'$與使用者 A 本身的帳號$ID_A$不會一致，使用者 A 會立即中斷與伺服器的連線，偽裝伺服器無法通過使用者 A 的身份驗證。在我們的通訊協定中，提供了伺服器 S 與使用者 A、B 之間的相互驗證，因此可以有效的避免伺服器偽裝攻擊(Server Spoofing Attack)。

4.5 使用者假冒攻擊(Impersonation Attack)

在我們的通訊協定中，使用者 A、B 以密碼的單向雜湊函數值$h(PW_A)$、$h(PW_B)$為對稱式金鑰，對g^x, ID_A, ID_B、g^y, ID_B, ID_A加密，得到$[g^x, ID_A, ID_B]_{h(PW_A)}$與$[g^y, ID_B, ID_A]_{h(PW_B)}$。當伺服器 S 要驗證使用者 A、B 的身份時，並非依據使用者的密碼PW_A, PW_B，或密碼的雜湊函數值$h(PW_A)$、$h(PW_B)$來做判斷；而是將使用者 A、B 傳送過來的帳號ID_A、ID_B，伺服器 S 以非對稱式私密金鑰k_S解密所得到的帳號ID_A、ID_B，以及$h(PW_A)$、$h(PW_B)$為對稱式金鑰解密所得到的帳號ID_A、ID_B做比對，根據使用者 A、B 這三個帳號是否完全相同，來驗證身份，確認要進行通訊的雙方確實為使用者 A、B。

假設有攻擊者擷取網路上所有通訊驗證的訊息，企圖假冒合法使用者的身份，登入伺服器，那麼攻擊者本身必須要能假造出一組合法使用者的帳號及密碼，才能通過伺服器的驗證。然而，在我們的通訊協定中，使用者 A、B 直接將密碼的雜湊函數值$h(PW_A)$、$h(PW_B)$，做為訊息加密的對稱式金鑰，加密後的訊息$[g^x, ID_A, ID_B]_{h(PW_A)}$與$[g^y, ID_B, ID_A]_{h(PW_B)}$，再以伺服器的非對稱式公開金鑰$K_S$加密保護後再傳送，攻擊者無法在網路上擷取到任何密碼PW_A, PW_B或密碼雜湊函數值$h(PW_A)$、$h(PW_B)$的訊息，所以無法假造合法使用者的驗證資料。如果攻擊者假造錯誤的$[g^x, ID_A, ID_B]_{h(PW_A')}$或$[g^y, ID_B, ID_A]_{h(PW_B')}$，在伺服器驗證通訊雙方的身份時，一定會被察覺，因此我們的通訊協定可以有效的避免使用者假冒攻擊(Impersonation Attack)。

4.6 重送攻擊(Replay Attack)

假設攻擊者在網路上擷取到使用者 A、B 與即時通訊伺服器 S 之間所有通訊驗證的

訊息，因為攻擊者沒有伺服器的非對稱式私密金鑰 k_S，所以無法取得 $[g^x, ID_A, ID_B]_{h(PW_A)}$、$[g^y, ID_B, ID_A]_{h(PW_B)}$、$r_A$、$r_B$ 等訊息。如果攻擊者直接將擷取到的 $ID_A, \{ID_A, r_A, [g^x, ID_A, ID_B]_{h(PW_A)}\}_{K_S}, g^x$ 與 $ID_B, \{ID_B, r_B, [g^y, ID_B, ID_A]_{h(PW_B)}\}_{K_S}, g^y$ 重新傳送給伺服器，企圖通過伺服器的驗證，計算使用者 A、B 的階段性共通金鑰。然而，當伺服器回應攻擊者的驗證要求，將 $[ID_A, r_S, g^y]_{r_A}$ 與 $[ID_B, r_S, g^x]_{r_B}$ 傳送給使用者 A、B 時，攻擊者沒有正確的 r_A 或 r_B，無法對 $[ID_A, r_S, g^y]_{r_A}$ 與 $[ID_B, r_S, g^x]_{r_B}$ 進行解密得到隨機亂數 r_S。同時根據解離散對數問題的困難性，攻擊者亦無法從 g^x 或 g^y 的值，計算出正確的 x 或 y 值。在我們的通訊協定中，$K = h(g^{xy}, r_S)$ 為使用者 A、B 的階段性共通金鑰(Session Key)，x, y, r_A, r_B, r_S 為使用者與伺服器隨機選擇的亂數，在每一次的驗證中均不相同，並且只使用一次，攻擊者無法取得 x, y, r_S 的值，計算出正確的階段性金鑰，因此我們的通訊協定可以有效的避免重送攻擊(Replay Attack)。

4.7 前推私密性(Forward Secrecy)

在我們的通訊協定中，$K = h(g^{xy}, r_S)$ 為使用者 A、B 交談的階段性共通金鑰(Session Key)，x, y, r_S 為使用者 A、B 與即時通訊伺服器 S 隨機選擇的亂數，在每一次的通訊驗證均不相同，因此每一次產生的階段性共通金鑰 $K = h(g^{xy}, r_S)$ 也不相同。在某一次的交談通訊結束後，用來對通訊訊息加解密的階段金鑰便失去作用，無法使用這一把金鑰解開先前交談加密傳送的訊息，也無法透過這一把金鑰來推算先前或之後通訊驗證所產生的階段金鑰。

假設使用者 A、B 與伺服器 S 在某一次通訊驗證挑選的隨機亂數 x, y, r_S 不慎洩漏，攻擊者能取得這些訊息，並且計算出該次的階段金鑰 $K = h(g^{xy}, r_S)$，然而這一把金鑰僅能解開該次交談加密的通訊訊息，當下一次進行通訊驗證時，使用者 A、B 與伺服器 S 所選擇的隨機亂數 x', y', r_S'，與前一次的隨機亂數 x, y, r_S 完全不同，所計算出來的階段金鑰 $K' = h(g^{x'y'}, r_S')$ 與前一次的階段金鑰 K 也不可能相同，攻擊者無法利用這一把階段金鑰 K，推算其他次的階段金鑰，進而解開先前交談加密過的通訊訊息，因此我們的通訊協定具備前推私密性(Forward Secrecy)。

4.8 效能分析

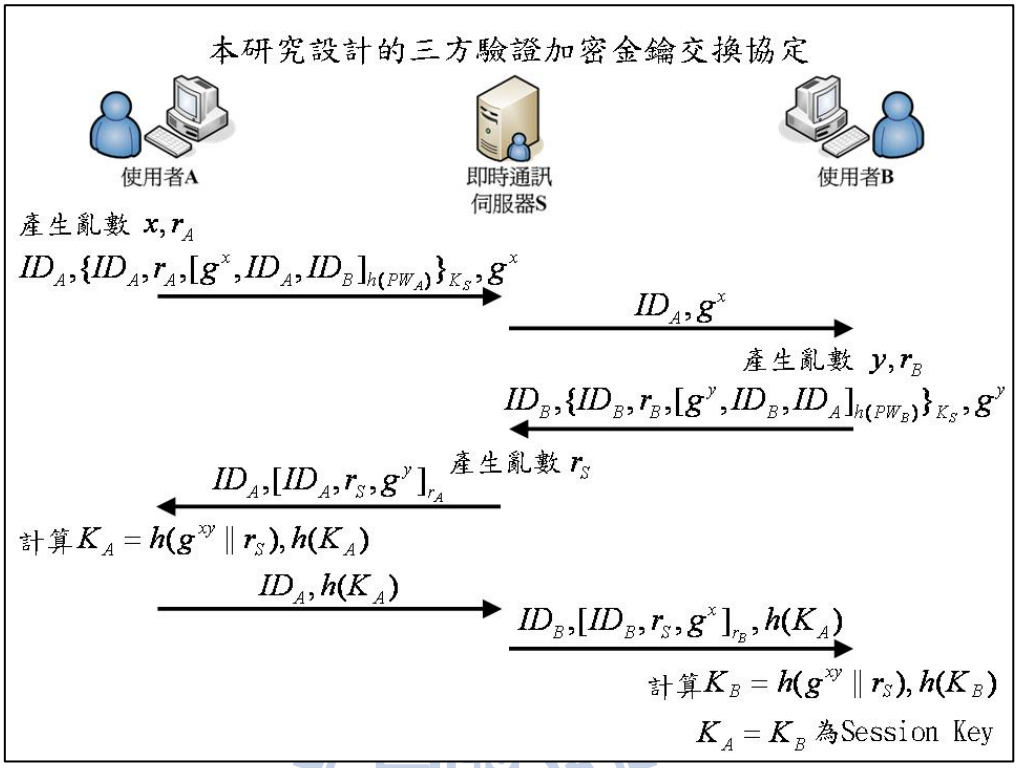

圖 4-1：本研究設計的三方驗證加密金鑰交換協定

如圖 4-1 所示，在我們的通訊協定中，當兩個使用者要開始交談，透過可信賴伺服器進行身份驗證與金鑰交換的流程，總共需傳送六次資料。在使用者端使用了兩個隨機亂數，進行兩次模指數運算、一次的對稱式金鑰加密、解密運算、一次的非對稱式公開金鑰加密運算，以及三次的雜湊函數運算；而在伺服器端使用了一個隨機亂數，進行二次的對稱式金鑰加密、解密運算，以及二次的非對稱式私密金鑰解密運算。使用者與伺服器的運算次數如下表：

表 4-1：運算次數表

	資料傳送次數	隨機亂數個數	指數運算次數	對稱加密次數	對稱解密次數	非對稱式加密次數	非對稱式解密次數	單向雜湊函數運算次數
伺服器 S	3	1	0	2	2	0	2	0
使用者 A	2	2	2	1	1	1	0	3
使用者 B	1	2	2	1	1	1	0	3

第五章 即時通訊系統實作與測試

本章就本研究設計的即時通訊協定進行實作，包含即時通訊伺服器與使用者的系統，並進行測試。

5.1 系統設計架構

本研究設計的即時通訊系統分為伺服器系統與使用者系統，各系統的功能規劃如下：

1.伺服器系統

伺服器的功能主要接受使用者的連線登入與驗證通訊要求，提供使用者知道其他聯絡人目前在即時通訊系統的線上狀態，並與線上的聯絡人驗證通訊。伺服器必須儲存每個使用者的身份帳號、驗證密碼，以及聯絡人清單，當使用者要求與聯絡人進行即時通訊時，伺服器協助雙方驗證彼此的身份，在使用者與聯絡人的身份皆驗證正確後，讓雙方建立一把階段性的共通金鑰。使用者與聯絡人使用這一把共通金鑰來對即時通訊的訊息加解密，防止訊息遭到擷取解密、窺視。

2.使用者系統

使用者需事先在伺服器完成註冊，將帳號與密碼經由安全通道傳送給伺服器，伺服器儲存這一組帳號、密碼。當使用者要登入即時通訊系統，使用伺服器的服務時，必須先連線至伺服器的網頁，下載 Java 的 Applet 程式，取得伺服器的公開金鑰，使用者與即時通訊伺服器連線後，傳送帳號至伺服器登入即時通訊系統。當使用者要與聯絡人進行通訊時，必須輸入身份驗證的密碼，並利用伺服器的公開金鑰對驗證資料加密後再傳送，在使用者、聯絡人與伺服器三方完成驗證的程序後，使用者與聯絡人計算出一把共通金鑰，利用這一把金鑰對即時通訊的訊息加解密。

5.2 系統開發工具

Java 為物件導向(Object Oriented)的程式語言，提供物件記憶體自動回收的管理機制(Garbage Collection)，以及許多的應用程式介面 Java API(Java Application Programming Interface)，包含網路通訊與密碼學演算法相關工具的類別函式，在網際網路的應用上，具備跨平台、高可攜性的優勢，因此我們選擇 Java 平台來實作即時通訊系統[20]。

由於伺服器必須提供內含 Java Applet 程式的網頁讓使用者下載，使用者透過 Java Applet 程式與伺服器連線，傳送帳號登入即時通訊系統，因此伺服器必須具備網頁伺服器(Web Server)的功能，並使用資料庫儲存使用者的帳號、密碼與聯絡人清單。基於這些功能考量，本研究以 Apache Web Server 作為網頁伺服器，並以 MySQL Database 作

為資料庫，儲存使用者的帳號相關資料。即時通訊系統的開發環境與工具，如下表：

表 5-1：即時通訊系統開發工具

用　　途	套件名稱與版本
作業系統	Microsoft Windows XP Professional Version 2002 Service Pack 3
Java 開發平台	Java(TM) SE Development Kit 6 Update 16 Windows x86 下載網址: http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u25-download-346242.html
AppServ 套件 (網頁伺服器及資料庫)	AppServ 2.4.9，含下列元件： 1.Apache Web Server Version 2.0.59 2.PHP Script Language Version 4.4.7 3.MySQL Database Version 5.0.45 4.phpMyAdmin Database Manager Version 2.10.2 下載網址: http://www.appservnetwork.com/
MySQL 連結 Java 平台 驅動程式	mysql-connector-java-5.0.8-bin.jar 下載網址: http://dev.mysql.com/downloads/connector/j/

5.3　密碼學演算法之選擇

Java 平台提供了許多密碼學工具的類別，根據 JCA(Java ™ Cryptography Architecture)Reference Guide 文件[3]與 Sun Providers Documentation[4]的說明，Java 平台提供了金鑰交換協定、對稱式加密、非對稱式加密、單向雜湊函數等密碼學技術的類別加密函式。在程式開發時，需要先 import 相關密碼學的類別庫，以及 BigInteger 類別，如圖 5-1 所示。

```
12  import java.math.BigInteger;
13  import java.security.*;
14  import java.security.spec.*;
15  import java.security.interfaces.*;
16  import javax.crypto.*;
17  import javax.crypto.spec.*;
18  import javax.crypto.interfaces.*;
19  import com.sun.crypto.provider.SunJCE;
20
21  public class MessengerServer
22  {
23    ServerSocket ss;
```

圖 5-1：import 相關密碼學的類別庫

基於解離散對數問題與大數分解的困難度，本研究設計的即時通訊協定以 Diffie-Hellman 金鑰交換協定為基礎，並且選擇 RSA 公開金鑰密碼系統做為伺服器的非對稱式加密金鑰。要在 Java 平台使用這兩種密碼學技術，需要先產生 Diffie-Hellman 金鑰交換協定的參數，以及 RSA 公開金鑰密碼系統的公開金鑰與私密金鑰，如圖 5-2 所示。

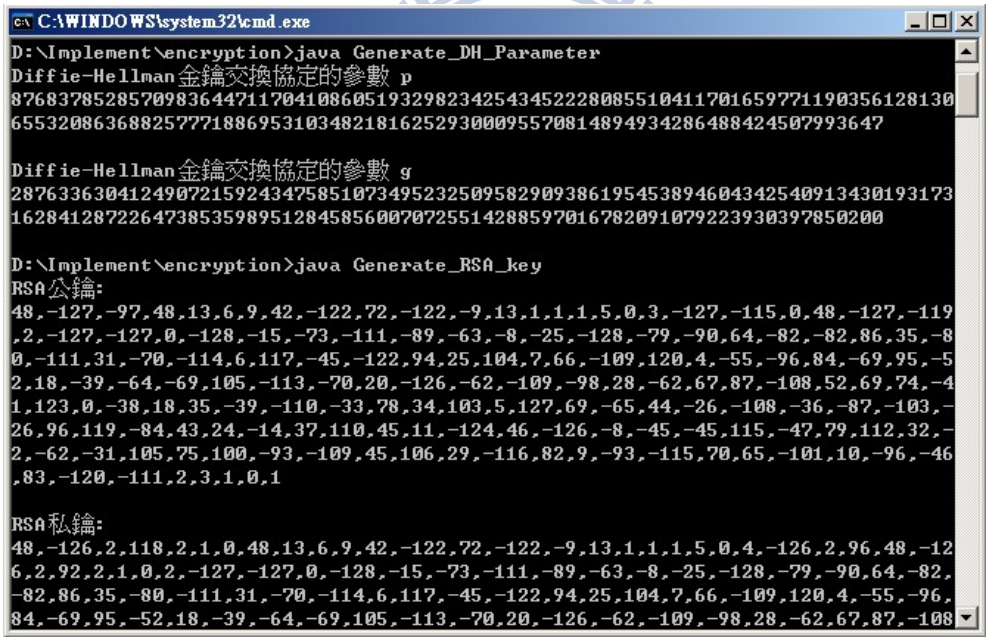

圖 5-2：產生 Diffie-Hellman 金鑰交換協定的參數及 RSA 金鑰對

由於 DES 的金鑰長度僅 56 位元，本研究採用金鑰長度 168 位元，安全性較高的 Triple-DES 三重資料加密標準，做為即時通訊協定的對稱式加密金鑰，單向雜湊函數則採用 SHA-1。本研究在 Java 平台所使用的密碼學類別主要由 SunJCE 所提供，詳細密碼學演算法的名稱與規格，如下表：

表 5-2：密碼學演算法規格

密碼學演算法	演算法名稱及規格
金鑰交換協定	Diffie-Hellman 公開金鑰交換演算法 參數長度 512 bits
非對稱式密碼系統	RSA 公開金鑰密碼系統 金鑰長度 1024 bits，ECB 模式
對稱式密碼系統	Triple-DES 三重資料加密標準 金鑰長度 168 bits，ECB 模式
單向雜湊函數	SHA-1 輸出長度 160 bits

5.4 即時通訊系統測試

使用者要使用即時通訊系統，必須先開啟瀏覽器，連線至 http://java.com/下載 Java 軟體(Java Runtime Environment，JRE)，並完成安裝。本研究使用者安裝的版本為 Java(TM) SE Runtime Environment 6 Update 25。

即時通訊系統使用者的介面設計如下，使用者要使用即時通訊系統的服務時，必須先連線至伺服器的網頁，下載網頁的 Applet 程式，如圖 5-3 所示。

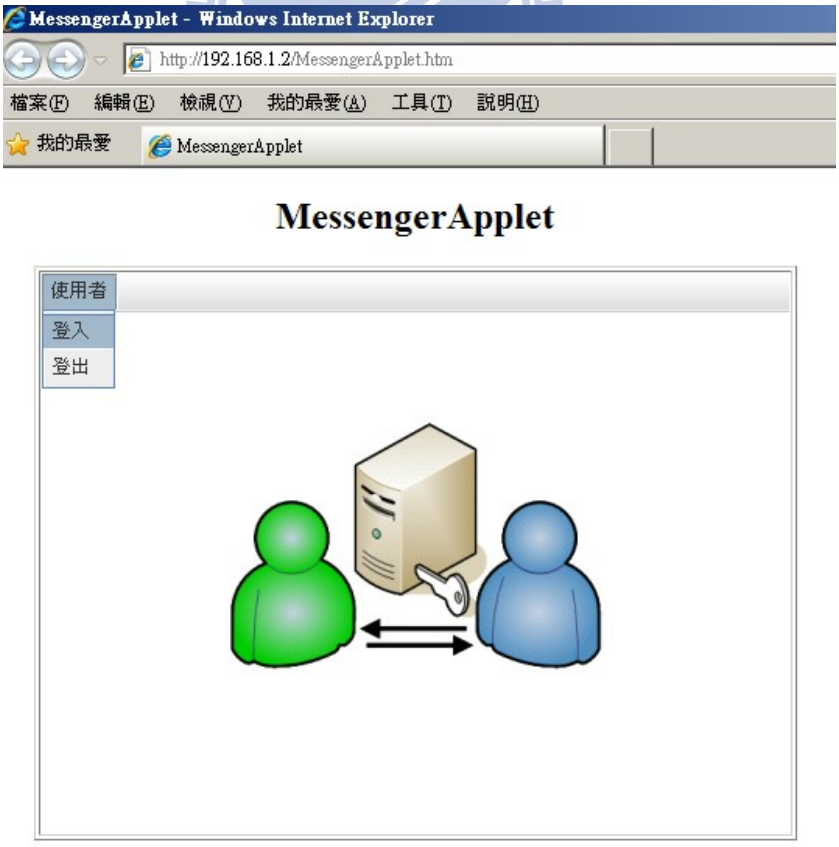

圖 5-3：即時通訊系統使用者介面

使用者透過網頁上的 Applet 程式與即時通訊伺服器連線，傳送帳號登入系統，即可使用即時通訊系統的服務，如圖 5-4 所示。

圖 5-4：使用者輸入帳號登入即時通訊系統

伺服器以 7777 為通訊埠開啟即時通訊服務，接受已經註冊的使用者以 Applet 程式連線登入系統，如圖 5-5 所示。

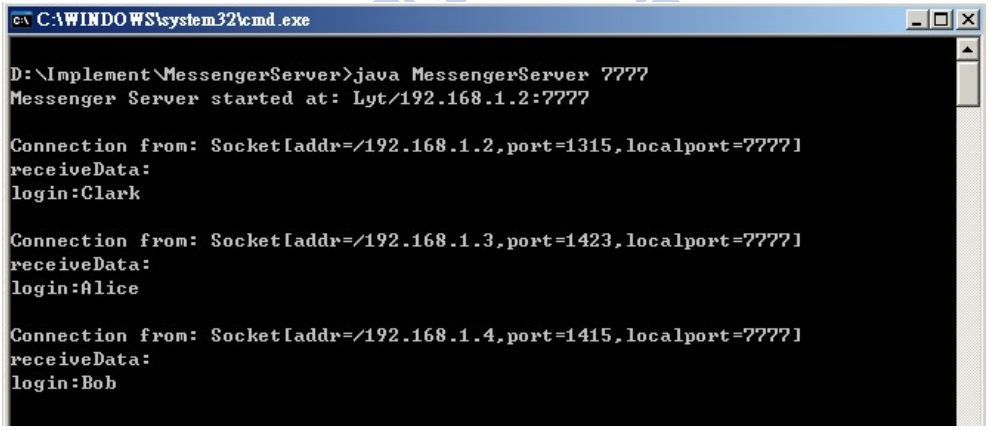

圖 5-5：即時通訊系統接受使用者連線登入

使用者以帳號成功登入系統後，即可得知目前有多少聯絡人登入伺服器，如圖 5-6 所示。

圖 5-6：使用者的聯絡人狀態

當使用者要與線上的聯絡人通訊時，Applet 程式會要求使用者輸入身份驗證的密碼，如圖 5-7 所示。

圖 5-7：使用者輸入身份驗證密碼

使用者輸入正確的密碼後，由 Applet 程式產生身份驗證的資料，傳送給即時通訊伺服器。伺服器驗證使用者的身份正確無誤後，再將通訊的要求轉發給聯絡人。聯絡人在收到要求通訊的訊息後，可以選擇是否與使用者通訊，如圖 5-8 所示。

圖 5-8：聯絡人選擇是否要與使用者通訊

若聯絡人接受使用者的通訊要求,Applet 程式同樣會要求聯絡人輸入身份驗證的密碼，如圖 5-9 所示。

圖 5-9：聯絡人輸入身份驗證密碼

伺服器使用資料庫儲存已註冊的使用者帳號與身份驗證密碼，當線上的使用者要與聯絡人通訊時，伺服器對雙方傳送過來的驗證資料進行解密，驗證使用者與聯絡人的身份，如圖 5-10 所示。

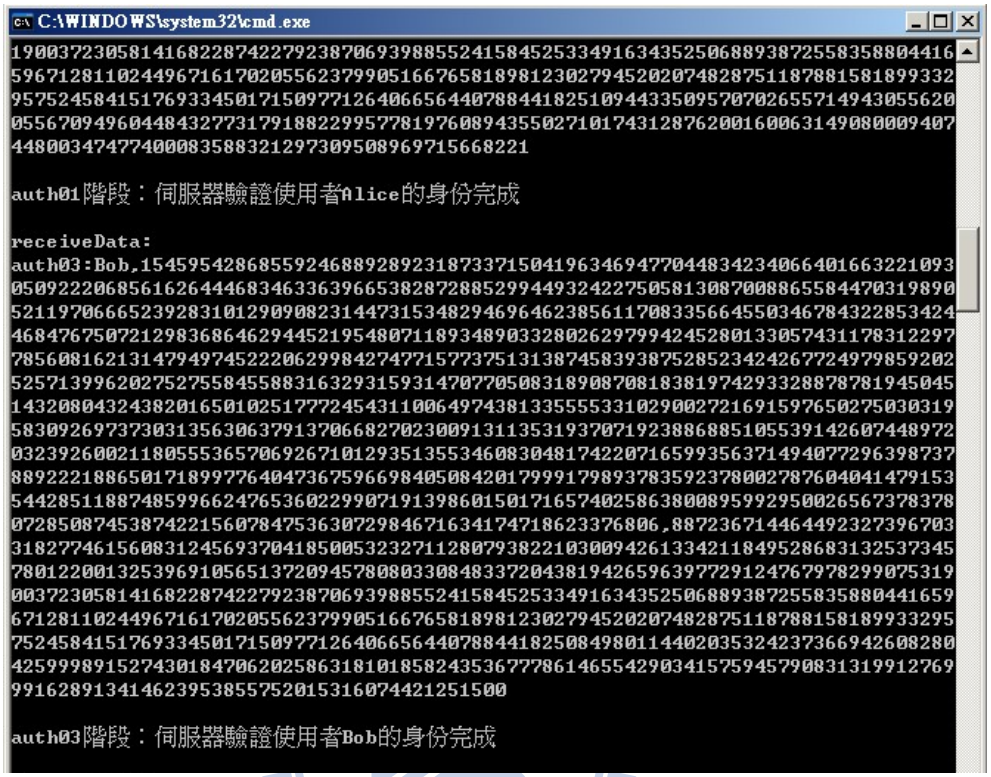

圖 5-10：伺服器驗證使用者與聯絡人的身份

使用者或聯絡人若以錯誤的身份驗證密碼來對驗證資料加密，伺服器在驗證身份解密的過程，會產生程式上的例外，使用者或聯絡人的連線將會被中斷，如圖 5-11 所示。

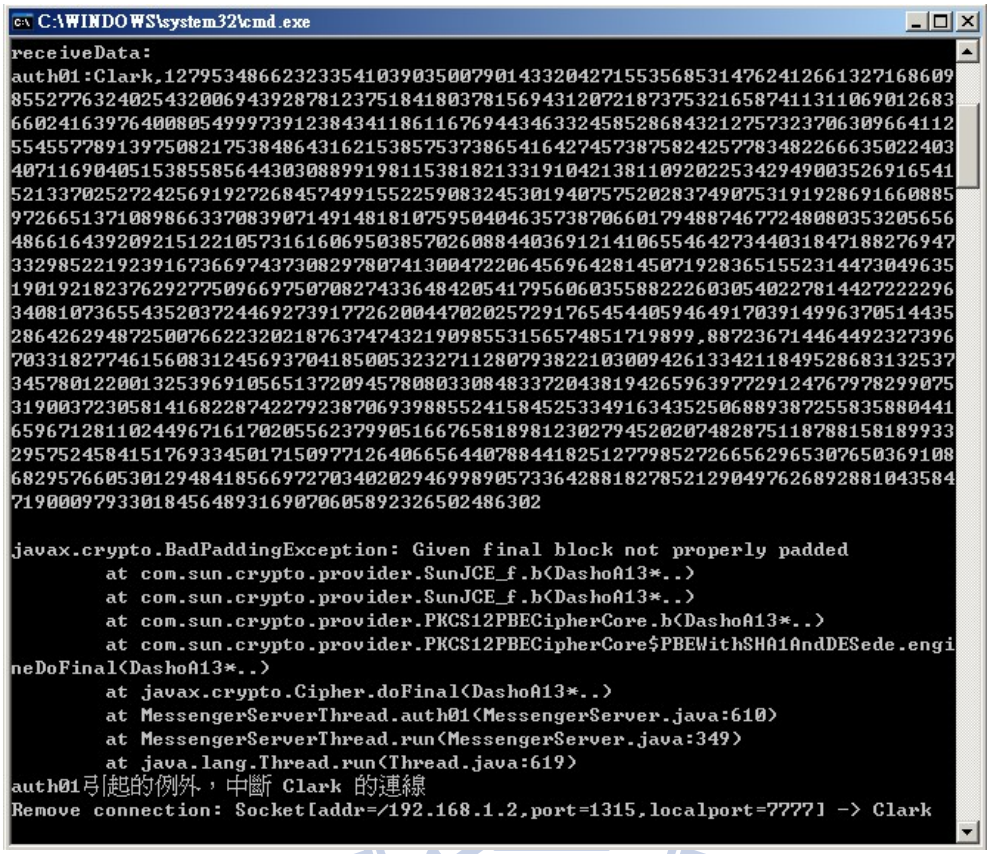

圖 5-11：伺服器中斷使用者的連線

在使用者、聯絡人與伺服器三方的身份驗證完成後，伺服器協助使用者與聯絡人建立一把階段性的共通金鑰。使用者使用這把共通金鑰對通訊的訊息加密，再傳送到伺服器，由伺服器轉發給聯絡人，聯絡人收到加密的訊息，便利用這一把共通金鑰來解密，還原成原來的訊息，如圖 5-12 所示。

圖 5-12：使用者與聯絡人通訊

第六章 結論與建議

本章提出研究成果結論與建議，以提供未來的研究方向。

6.1 結論

隨著網際網路的發展普及，即時通訊軟體提供人們即時的訊息傳送，為生活帶來許多的便利性，有越來越多的人把即時通訊軟體作為日常生活聯絡溝通的工具。然而，隨著即時通訊軟體的普遍使用，隨之而來的安全問題也逐漸浮現，如何在網路上提供使用者安全的即時通訊環境，是一項日益重要的安全議題。

本研究以學者 Lee 等人所提出的三方驗證金鑰交換協定為研究基礎，就理論與現實網路環境不合理的部分提出改善，設計安全的即時通訊協定，並採用 Diffie-Hellman 公開金鑰交換演算法、RSA 公開金鑰密碼系統、Triple-DES 三重資料加密標準與 SHA-1 單向雜湊函數等密碼學技術，在 Java 平台實作即時通訊系統。我們的協定讓伺服器與使用者之間能相互驗證彼此的身份，並在通訊的兩位使用者建立一把階段性的共通金鑰。即時通訊系統能符合驗證性、機密性、完整性、前推私密性等安全需求，避免目前常見的網路攻擊方法。

6.2 建議

本研究的即時通訊系統提供兩位使用者間文字內容的傳輸，未來可以在通訊協定上改良，朝向多方通訊發展，並加入檔案、視訊、音訊等多媒體串流傳輸的功能。此外，在本研究的即時通訊系統，使用者以伺服器的非對稱式公開金鑰來對驗證的資料加密傳送；然而，採用公開金鑰密碼系統，便有公鑰認證管理的問題，未來可以在系統加入智慧卡的使用，以避免網路釣魚類型的攻擊，同時改善在遠端伺服器儲存使用者驗證表的缺點，提升系統的安全性。

參考文獻

[1] H.T. Yeh, H.M. Sun, T. Hwang, "Efficient three-party authentication and key agreement protocols resistant to password guessing attacks", Information Science and Engineering, Vol.19, No.6, pp.1059-1070, 2003.

[2] M. Day, J. Rosenberg, H. Sugano, "A Model for Presence and Instant Messaging", IETF RFC 2778, February 2000, from World Wide Web: http://www.ietf.org/rfc/rfc2778.txt

[3] Oracle and/or its affiliates., Java ™ Cryptography Architecture (JCA) Reference Guide for Java ™ Platform Standard Edition 6, 2011, from World Wide Web: http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

[4] Oracle and/or its affiliates., Java ™ Cryptography Architecture Sun Providers Documentation for Java™ Platform Standard Edition 6, 2011, from World Wide Web: http://download.oracle.com/javase/6/docs/technotes/guides/security/SunProviders.html

[5] R.L. Rivest, A. Shamir, L.M. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Comm. Of the ACM, Vol.21, pp.120-126, 1978.

[6] Stallings, W., Cryptography and network security: Principles and Practice, 3rd Edition, Prentice Hall, Inc., 2003.

[7] T.F. Lee, J.L. Liu, M.J. Sung, S.B. Yang, C.M. Chen, "Communication-efficient three-party protocols for authentication and key agreement", Computers and Mathematics with Applications, 58, pp.641-648, 2009.

[8] W. Diffie, M.E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Vol.IT-22, No.6, pp.644-654, 1976.

[9] Y. Ding, P. Horster, "Undetectable on-line password guessing attacks", ACM Operating System Review, Vol.29, No.4, pp.77-86, 1995.

[10] 宋志民，陳大任，潘杏惠，ＭＳＮ密碼被盜 駭遍演藝圈經紀人-Yahoo!奇摩新聞，存取於 2010 年 3 月 31 日，http://tw.news.yahoo.com/article/url/d/a/100331/4/231a9.html。

[11] 李文輝，「一個有效率的三方金鑰交換協定」，朝陽科技大學，碩士論文，民國 98 年。

[12] 李尚宸，「使用通行碼之數位簽章協定」，國立交通大學，碩士論文，民國 92 年。

[13] 周伯錕，「利用智慧卡之遠端身份認證之研究」，國立中興大學，碩士論文，民國 92 年。
[14] 邱玉忠，「應用智慧卡之使用者確認機制之研究」，國立南台科技大學，碩士論文，民國 94 年。
[15] 張真誠，林祝興，資訊安全技術與應用，全華科技圖書股份有限公司，初版，台北，民國 95 年。
[16] 張雅芬，「身份認證與數位簽章技術及其在電子商務上的應用」，國立中正大學，博士論文，民國 94 年。
[17] 許行，「具高效率前推私密性之安全電子郵件協定」，國立中山大學，碩士論文，民國 96 年。
[18] 陳帥名，「免儲存驗證資訊之通行碼身份認證協定」，輔仁大學，碩士論文，民國 93 年。
[19] 黃雅伶，「MifareCard 結合 agsXMPP 建構安全的即時通訊軟體」，國立高雄師範大學，碩士論文，民國 96 年。
[20] 黃嘉輝，Java 網路程式設計，文魁資訊股份有限公司，初版，台北市，民國 95 年。
[21] 楊中皇，郭宗益，「基於橢圓曲線密碼學的安全即時通訊系統設計與實現」，第十七屆資訊安全會議，136~145 頁，嘉義，民國 96 年 6 月 7 日。
[22] 楊中皇，網路安全理論與實務，學貫行銷股份有限公司，初版，台北，民國 97 年。
[23] 楊佑寧，「有限信任讀卡機下安全服務機制」，國立暨南大學，碩士論文，民國 95 年。
[24] 路怡珍，究竟誰上線？ MSN 詐騙「好友」下手-Yahoo!奇摩新聞，存取於 2010 年 5 月 8 日，http://tw.news.yahoo.com/article/url/d/a/100507/8/259xu.html。
[25] 蔡佳倫，「遠端使用者身份驗證之研究」，國立交通大學，碩士論文，民國 96 年。
[26] 賴溪松，韓亮，張真誠，近代密碼學及其應用，初版，旗標出版股份有限公司，台北，民國 92 年。
[27] 謝續平，2010-春-網路安全上課講義，存取於 2010 年 3 月 20 日，http://dsns.csie.nctu.edu.tw/blog/index.php?option=com_agora&task=topic&id=61&Itemid=126，2010。